

ВОЛОГОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

А. Водовозов

ЦИФРОВЫЕ ЭЛЕМЕНТЫ СИСТЕМ АВТОМАТИКИ

Учебное пособие

Допущено учебно-методическим объединением по образованию в области энергетики и электротехники в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальности 180400 «Электропривод и автоматика промышленных установок и технологических комплексов» направления 654500 «Электротехника, электромеханика и электротехнологии»

Вологда
2002

УДК 621.383
ББК 32.85
В62

Рецензенты:

Кафедра робототехники и автоматизации производственных систем Санкт-Петербургского государственного технического университета, зав. кафедрой профессор
Рассудов Л.Н

Зав. кафедрой автоматизированного электропривода Комсомольского на Амуре государственного технического университета, профессор Соловьев В.А.

Водовозов А.М. Цифровые элементы систем автоматики. Учебное пособие. - Вологда. ВоГТУ, 2001.- 108 с., ил.

Рассматриваются теоретические основы цифровой автоматики и типовые компоненты цифровых систем управления: элементы стандартной логики, микросхемы памяти, большие интегральные схемы с программируемой структурой, аналого-цифровые и цифроаналоговые преобразователи, цифровые датчики электрических и технологических параметров.

Допущено учебно-методическим объединением по образованию в области энергетики и электротехники в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальности 180400 «Электропривод и автоматика промышленных установок и технологических комплексов» направления 654500 «Электротехника, электромеханика и электротехнологии».

Пособие может быть полезно инженерно-техническим работникам и студентам различных специальностей, изучающим цифровую схемотехнику.

ISBN

ISBN

© Водовозов А. М., 2002
© Вологодский государственный технический университет, 2002

ВВЕДЕНИЕ

Вопросы цифровой схемотехники исторически рассматриваются как обязательная часть образовательной программы подготовки инженеров по электроприводу и автоматике. Эти вопросы всегда вызывают повышенный интерес студентов, пытающихся проникнуть в тайны цифровых регуляторов и элементов цифровых систем управления. Предлагаемая книга задумана как учебное пособие по элементам цифровой схемотехники и, одновременно, как краткий справочник-путеводитель по современной элементной базе цифровых устройств. В ней в максимально упрощенной форме рассмотрены все аспекты развития этого направления современной автоматики.

Основой для написания книги послужили конспекты учебного курса, читаемого автором студентам специальности 180400 на протяжении более чем 20 лет. За этот период цифровая схемотехника, которую раньше рассматривали как раздел промышленной электроники, превратилась в довольно абстрактную науку. Сейчас она оперирует своими логическими понятиями, терминами и определениями. Базируясь на сравнительно простом математическом аппарате булевой алгебры, она позволяет решать задачи синтеза сложных высокоинтеллектуальных технических устройств формальными методами, опираясь на множество современных интегральных схем, созданных ведущими мировыми производителями. Именно в такой постановке рассматривается материал в данной книге. В ней рассматриваются ключевые понятия и модели, практические схемы и микросхемы отечественных и зарубежных производителей, решающие самые различные задачи. В книге собран материал, позволяющий освоить курс студентам с разной базовой подготовкой. Отказ от глубокой математической и физической трактовки процессов объясняется стремлением максимально приблизить обучаемого к реалиям современной схемотехники, показать ему возможности этой техники и пути её развития.

В первых главах учебника кратко рассмотрены математические основы цифровой схемотехники: двоичная арифметика и алгебра логики. Современные полупроводниковые элементы, составляющие основу цифровой схемотехники, разобраны в третьей главе. Основные технологии производства интегральных схем, созданные на различных этапах электроники, представлены в четвертой главе работы. Все последующие главы книги посвящены интегральной схемотехнике в её сегодняшнем представлении, её терминам и разработкам, классическим и современным схемам.

Книга не претендует на роль справочника. Приведенные в ней в качестве примеров различные интегральные схемы скорее можно отнести к простейшим. Они не могут даже частично отразить существующего многообразия изделий, созданного известными всем гигантами электронного бизнеса: *Motorola, Siemens, Sony, Samsung, Micron, Toshiba, Texas Instruments, NEC, Intel, AMD.....* Уже сегодня размер транзистора на кристалле полупроводника измеряется долями микрона, количество транзисторов – десятками миллионов, рабочая частота цифровых микросхем – сотнями мегагерц. Развитие схемотехники настолько стремительно, что охватить всё многообразие существующих схем в таком пособии просто невозможно.

1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ АВТОМАТИКИ

1.1. Информация и общие принципы ее преобразования

Понятие информации является важнейшим для современной автоматики. Процессы её обработки, приема, передачи и хранения лежат в основе функционирования

всех цифровых устройств. При этом под информацией понимаются систематизированные сведения о тех или иных объектах, сведения об определенных свойствах или параметрах этих объектов и о зависимостях между этими свойствами.

Информация воплощенная и зафиксированная в некоторой материальной форме, называется *сообщением*. Средством передачи сообщения является *сигнал*, под которым понимается процесс изменения во времени некоторого физического параметра $x(t)$. Процесс обычно является непрерывным (аналоговым) и описывается соответствующими графиками или аналитическими выражениями. При этом имеется в виду, что причиной изменения величины x является не само время, а некоторая другая физическая величина.

При цифровой обработке информации осуществляются преобразования аналоговых сигналов: дискретизация, квантование и кодирование.

Дискретизация

Дискретизацией называется выборка мгновенных значений (отсчетов) непрерывного сигнала с равными интервалами времени Δt между ними. Полученный в результате такого преобразования *дискретный сигнал* описывается решетчатой функцией (последовательностью) $x(n\Delta t) \equiv x(n)$, в которой: Δt - интервал (шаг) дискретизации, а независимая переменная n принимает только дискретные значения $n = 0, 1, 2, \dots$. (рис. 1.1).

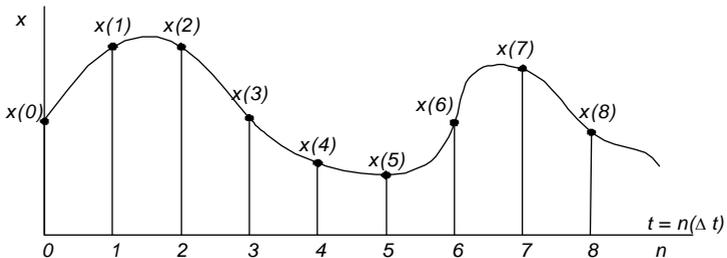


Рис. 1.1. Дискретизация непрерывного сигнала

В результате дискретизации сигнал $x(t)$ заменяется совокупностью отдельных значений $x(n)$.

Величина $f = 1/\Delta t$ называется частотой дискретизации. Правильный выбор частоты дискретизации устанавливает теорема Котельникова, в соответствии с которой для полного восстановления непрерывного сигнала по его дискретным отсчетам необходимо, чтобы частота дискретизации f как минимум вдвое превышала наибольшую частоту спектра аналогового сигнала.

Квантование

При квантовании значение сигнала $x(n)$ заменяется ближайшим, из определенного заранее множества, дискретным значением. В результате функция приобретает ступенчатый вид (рис.1.2.):

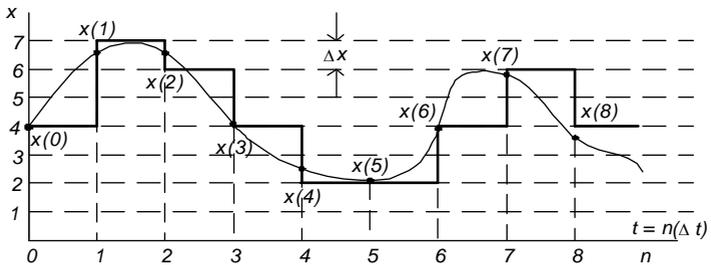


Рис. 1.2. Квантование сигнала

Шагом квантования сигнала x называется разность между соседними дискретными значениями функции. Обычно шаг квантования выбирается постоянным и равен : $\Delta x = x_{\max} / (N - 1)$, где N - число уровней квантования.

Абсолютная погрешность квантования по уровню может изменяться от 0 до $\Delta x / 2$.

Кодирование

При кодировании каждому значению квантованного сигнала ставится в соответствие некоторое число (код) из конечного множества чисел, определенного числом уровней квантования.

В результате цифровой сигнал описывается квантованной решетчатой функцией (квантованной последовательностью) $x(n)$, которая принимает ряд дискретных значений уровней квантования $x(0)$, $x(1)$, $x(2)$, каждому из которых поставлено в соответствие некоторое число. Поэтому передача или обработка отсчета цифрового кодированного сигнала сводится к операциям над безразмерными числами (кодами). Представление сигнала в цифровой форме практически всегда дает существенное преимущество при передаче, хранении и обработке информации.

Дискретизация, квантование и кодирование электрических аналоговых сигналов в автоматике обычно осуществляется при помощи специализированных схем аналого-цифровых преобразователей (*Analog-Digital Converter*).

Мера информации

Для оценки количества информации в цифровом сигнале используются различные подходы.

Наиболее распространена в настоящее время аддитивная мера информации (мера Хартли), в соответствии с которой количество информации измеряется в двоичных единицах - битах.

Для любого числа вводятся понятия:

- q – количество цифр, используемых для представления сигнала;
- n – разрядность чисел, достаточная для представления всех значений квантованного сигнала.

При заданной количестве цифр и разрядности чисел общее количество всевозможных уровней квантования равно $N = q^n$. А для оценки количества информации

вводится логарифмическая мера – бит: $I = \log_2 N = n \log_2 q$. Полученный результат округляется вверх до ближайшего целого числа.

Например, если сигнал описывается двухразрядными десятичными числами, то количество цифр $q = 10$, а количество разрядов числа $n = 2$. В результате: $I = 2 \cdot \log_2 10 \approx 7$. Т.е. одно дискретное квантованное значение сигнала несет в себе 7 бит информации.

При наличии нескольких источников информации общее её количество равно сумме $I = I_1 + I_2 + \dots$, где I_i - количество информации от i -го источника.

1.2. Системы счисления

В процессе переработки информации автоматы оперируют цифрами. Каждой цифре числа однозначно сопоставляется ее количественный (числовой) эквивалент.

Совокупность приемов и правил, установленных для записи чисел с помощью цифр, устанавливается системой счисления. Запись числа в некоторой системе счисления называют кодом числа.

Различают позиционные и непозиционные системы счисления.

Непозиционная система счисления - это система, для которой значение символа, т.е. цифры, не зависит от его положения в числе. К таким системам относится, в частности, римская система (правда с некоторыми оговорками). Здесь, например, символ V всегда означает пять, вне зависимости от места его появления в записи числа. Есть и другие современные непозиционные системы.

Позиционная система счисления - это система, в которой значение каждой цифры зависит от ее числового эквивалента и от ее места (позиции) в числе, т.е. одна и та же цифра может иметь различные значения. Именно такие системы счисления в основном используются сейчас в повседневной жизни, в автоматике и вычислительной технике. Поэтому в дальнейшем будем рассматривать только позиционные системы.

Любая позиционная система счисления характеризуется основанием. Основание q позиционной системы счисления это количество знаков или символов, используемых в данной системе для изображения числа. В десятичной системе используется десять цифр от 0 до 9, основание $q = 10$; в двоичной системе две цифры 0 и 1, основание $q = 2$. Возможно бесчисленное множество позиционных систем, т.к. за основание можно принять любое число, образовав новую систему счисления.

При записи некоторого числа в позиционной системе счисления соответствующие цифры числа размещаются по отдельным нужным позициям, которые принято называть разрядами числа в данной позиционной системе счисления. Количество разрядов в записи числа называется разрядностью числа.

В общем случае в системе с основанием q число A представляется в виде полинома:

$$A = a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \dots + a_1q^1 + a_0q^0 + a_{-1} \cdot q^{-1} + a_{-2} \cdot q^{-2} \dots, \quad (1.1)$$

где n - количество цифр в целой части числа, а в качестве коэффициентов a_i могут стоять любые из множества q цифр, используемых в системе.

Числа в позиционной системе представляются в виде последовательности соответствующих цифр: $A = a_{n-1} \dots a_2 a_1 a_0, a_{-1} a_{-2} \dots$. Запятая отделяет целую часть числа от дробной (коэффициенты при положительных степенях, включая нуль, от коэффициентов при отрицательных степенях). Запятая опускается, если нет отрицательных степеней. При

такой записи вес каждого разряда увеличивается в q раз, по сравнению с соседним, при движении справа налево. Нулевым считается разряд справа от запятой.

Основной в автоматике считается двоичная система счисления. Наряду с ней активно используются системы счисления с основаниями 8, 16 и 10.

Двоичные числа

Основанием двоичной системы счисления является число 2 и в записи вида (1.1) используется только две цифры $a_i = 0$ или 1. Поэтому полином (1.1) выглядит следующим образом:

$$A = a_{n-1}2^{n-1} + \dots + a_22^2 + a_12^1 + a_02^0 + a_{-1}2^{-1} + a_{-2}2^{-2} + \dots \quad (1.2)$$

В тех случаях, когда в одном документе используются одновременно разные системы счисления, двоичные числа обычно обозначают символами b или B , записываемыми слева от числа. Например, по формуле (1.2):

$$b11010,01 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 26,25$$

Количество информации, содержащееся в n -разрядном двоичном числе равно количеству разрядов этого числа: $I = n \cdot \log_2 2 = n$ (бит). Поэтому отдельные разряды двоичных чисел принято называть битами.

Восьмеричные числа

В восьмеричной системе используется восемь цифр: 0, 1, 2, 3, 4, 5, 6, 7. Запись вида (1.1) в этом случае соответствует полиному:

$$A = a_{n-1}8^{n-1} + \dots + a_28^2 + a_18^1 + a_08^0 + a_{-1}8^{-1} + a_{-2}8^{-1} \dots \quad (1.3)$$

Восьмеричные числа принято отмечать слева символами O, Q или q. Например:

$$q4712,2 = 4 \times 8^3 + 7 \times 8^2 + 1 \times 8 + 2 \times 8^0 + 2 \times 8^{-1} = 2506,25 .$$

Количество информации, содержащееся в n -разрядном восьмеричном числе:

$$I = n \cdot \log_2 8 = 3n \text{ (бит)} .$$

Шестнадцатиричные числа

В шестнадцатиричной системе счисления используется 16 цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Запись шестнадцатиричного числа представляется в виде полинома с основанием 16:

$$A = a_{n-1}16^{n-1} + \dots + a_216^2 + a_116^1 + a_016^0 + a_{-1}16^{-1} + a_{-2}16^{-2} + \dots \quad (1.4)$$

Шестнадцатиричные числа обычно обозначаются слева символами H, h, X или x. Например:

$$hB2E,4 = B \times 16^2 + 2 \times 16^1 + E \times 16^0 + 4 \times 16^{-1} = 2862,25 .$$

Количество информации, содержащееся в n -разрядном шестнадцатиричном числе:

$$I = n \cdot \log_2 16 = 4n \text{ (бит)} .$$

В таблице 1.1. десятичные целые числа от 0 до 19 представлены в различных системах счисления.

Таблица 1.1.

Представление чисел в различных системах счисления

Десятичная	0	1	2	3	4	5	6	7	8	9
Двоичная	b0	b1	b10	b11	b100	b101	b110	b111	b1000	b1001
Восьмеричная	q0	q1	q2	q3	q4	q5	q6	q7	q10	q11
Шестнадцатеричная	h0	h1	h3	h3	h4	h5	h6	h7	h8	h9
Десятичная	10	11	12	13	14	15	16	17	18	19
Двоичная	b1010	b1011	b1100	b1101	b1110	b1111	b10000	b10001	b10010	b10011
Восьмеричная	q12	q13	q14	q15	q16	q17	q20	q21	q22	q23
Шестнадцатеричная	hA	hB	hC	hD	hE	hF	h10	h11	h12	h13

Если двоичная система счисления считается основной в цифровой автоматике, то восьмеричная и шестнадцатеричная обычно трактуются как удобный способ представления двоичных чисел.

В частности, для перехода от двоичной к восьмеричной записи двоичное число, начиная с запятой, разбивают на группы по 3 символа. Если в последней от запятой группе остается менее трех символов, то её снаружи дополняют нулями. Затем каждую группу заменяют соответствующим восьмеричным символом в соответствии с табл. 1.1. Например: b001 101 111 110, 001 = q1576,1.

Переход от двоичной к шестнадцатеричной системе аналогичен, но группы состоят из четырех символов: b1111 1000 1011 1110, 1000 = h7C5F,8.

1.3. Формальные правила двоичной арифметики

Все математические преобразования в цифровых схемах выполняются с использованием двоичной арифметики. При выполнении арифметических операций с десятичными, восьмеричными и шестнадцатеричными числами они предварительно преобразуются в двоичный код. Основными математическими операциями считаются: сложение, вычитание, умножение и деление переменных.

Сложение

Двоичное сложение выполняется в соответствии с таблицей сложения (табл. 1.2):

Таблица 1.2

Таблица двоичного сложения

0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	1 0

В последней строке таблицы в результате сложения двух цифр получается двухразрядное число, т.е. возникает перенос в старший разряд.

Сложение чисел начинается с младших разрядов. Возникающий перенос учитывается при сложении следующих (старших) разрядов. Например:

$$\begin{array}{r}
 1\ 0\ 1, \overset{\vee}{0}\ 1 \quad \text{слагаемые} \\
 + \underline{0\ 1\ 0, 0\ 1} \\
 1\ 1\ 1, 1\ 0 \quad \text{сумма}
 \end{array}$$

Здесь символом « \vee » отмечен перенос из младшего разряда суммы.

Вычитание

Таблица для операции вычитания двоичных цифр аналогична таблице сложения (табл. 1.3):

Таблица 1.3

Таблица двоичного вычитания

0	-	0	=	0
1	-	0	=	1
1	-	1	=	0
0	-	1	=	1 1

В последней строке таблицы в результате вычитания цифр формируется двухразрядное число, т.е. возникает заем из старшего разряда.

Вычитание чисел начинается с младших разрядов. Возникающий заем учитывается при вычитании следующих (старших) разрядов. Например:

$$\begin{array}{r}
 1\ 0\ 1, \overset{\vee}{0}\ 1 \quad \text{уменьшаемое} \\
 - \underline{0\ 1\ 0, 0\ 1} \quad \text{вычитаемое} \\
 0\ 1\ 1, 0\ 0 \quad \text{разность}
 \end{array}$$

Здесь символом « \vee » отмечен заем из старшего разряда.

Умножение

Умножение двоичных цифр производится в соответствии с таблицей умножения (табл. 1.4):

Таблица 1.4

Таблица двоичного умножения

0	x	0	=	0
0	x	1	=	0
1	x	0	=	0
1	x	1	=	1

Умножение двоичных чисел выполняется последовательно путем вычисления частичных произведений и их последующего суммирования. В соответствии с таблицей умножения каждое частичное произведение равно 0, если в соответствующем разряде множителя стоит 0, или равно множимому, сдвинутому на соответствующее число разрядов влево, если в разряде стоит 1. При суммировании частичных произведений получается результат умножения. Например:

$$\begin{array}{r}
 1\ 0\ 1 \quad \text{множимое} \\
 \times \underline{1\ 0\ 0\ 1} \quad \text{множитель} \\
 1\ 0\ 1 \\
 0\ 0\ 0 \\
 0\ 0\ 0 \\
 \hline
 1\ 0\ 1\ 0\ 0\ 1 \quad \text{частичные произведения}
 \end{array}$$

$$\begin{array}{r} 101 \\ \hline 101101 \end{array} \quad \text{результат}$$

Умножение связано с выполнением нескольких операций сложения и сдвига и, поэтому, требует существенно большего времени для вычисления, чем сложение и вычитание. Под результат двоичного умножения должно быть зарезервировано в два раза больше бит, чем под каждый из сомножителей.

Деление

Двоичное деление выполняется с помощью операций вычитания и сдвига. Делитель помещается под делимым со стороны старших разрядов (как можно левее) и производится вычитание. В старший разряд частного заносится единица. Затем делитель сдвигается на один разряд вправо и вновь осуществляется вычитание. Если вычитание невозможно из-за того, что остаток оказывается отрицательным, оно не выполняется и в соответствующий разряд частного заносится 0. Операция заканчивается при достижении последнего разряда делимого. Например:

$$\begin{array}{r} \text{делимое} \qquad \qquad \qquad \text{делитель} \\ 1011101 \mid 101 \\ \underline{101} \qquad \qquad \qquad 1001 \\ \qquad 0101 \qquad \qquad \text{частное} \\ \qquad \underline{0101} \\ \qquad \qquad 0000 \\ \qquad \qquad \qquad \text{остаток} \end{array}$$

1.4. Числа с фиксированной запятой

Для записи двоичного числа в цифровой схеме обычно выделяется определенное количество бит (разрядная сетка). Простейшие устройства обычно выполняются четырех или восьмибитными. Более сложные схемы имеют 16, 32 и более разрядов. При этом положение запятой фиксируется в определенном месте разрядной сетки. Обычно подразумевается, что запятая находится или перед старшим разрядом или после младшего. В первом случае могут быть представлены числа, которые по модулю меньше 1, во втором – только целые числа.

Для кодирования знака обычно предусматривается специальный «знаковый» разряд. В этом разряде 0 соответствует плюсу, а 1 – минусу.

На рис. 1.3 показана 8-разрядная сетка, где запятая зафиксирована после младшего разряда. Веса отдельных разрядов в этом случае меняются от 2^0 до 2^6 .

	7	6	5	4	3	2	1	0
	знак	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Рис. 1.3. Представление целых чисел с фиксированной запятой

В 8-разрядной сетке могут быть представлены целые числа от $b01111111 = +2^7 - 1 = 127$ до $b11111111 = -(2^7 - 1) = -127$.

На рис. 1.4 изображена 8-разрядная сетка, где запятая зафиксирована перед старшим разрядом. Веса разрядов в этом случае меняются от 2^{-7} до 2^{-1} .

	7	6	5	4	3	2	1	0
знак	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	

Рис. 1.4. Представление дробных чисел с фиксированной запятой

В такой разрядной сетке можно представить числа в диапазоне от $b01111111$ до $b11111111 = \pm (1 - 2^{-7}) = 0,9921875$. Минимальное по абсолютной величине дробное число, которое можно записать в восьмиразрядной сетке, равно $b00000001$ или $b10000001$ равно $\pm 2^{-7} = \pm 1/128 = \pm 0,0078125$.

При вычислениях необходимо, чтобы все исходные и промежуточные результаты не выходили за диапазон чисел, представимых в данной разрядной сетке. Для этого данные обычно берутся с масштабными коэффициентами.

1.5. Отрицательные числа

Рассмотренное выше представление двоичных чисел со знаком, где для записи знака в разрядной сетке отводится старший разряд, принято называть прямым кодом.

В общем случае прямой код n - разрядного двоичного числа со знаком имеет вид:

$$A = a_{n-1}a_{n-2}a_{n-1} \dots a_1a_0$$

$$\text{где: } a_{n-1} = \begin{cases} 0, & \text{при } A \geq 0, \\ 1, & \text{при } A < 0. \end{cases}$$

Если рассматривать прямой код, как некоторое число без знака, то вес знакового разряда a_{n-1} у целых чисел (рис. 1.3) равен 2^{n-1} , а у дробных (рис. 1.4) - $2^0 = 1$.

Изображение целого числа $[A]_{пр}$ в прямом коде в разрядной сетке автомата можно описать следующим образом:

$$[A]_{пр} = \begin{cases} A, & \text{при } A \geq 0, \\ 2^{n-1} + |A|, & \text{при } A < 0. \end{cases} \quad (1.5)$$

Аналогичным образом представляется изображение правильной дроби:

$$[A]_{пр} = \begin{cases} A, & \text{при } A \geq 0, \\ 1 + |A|, & \text{при } A < 0. \end{cases} \quad (1.6)$$

При использовании прямого кода сложение чисел, имеющих одинаковые знаки, выполняется по обычным правилам двоичной арифметики: числа складываются, сумме присваивается код знака слагаемых. Однако операция алгебраического сложения чисел с разными знаками существенно затрудняется. В этом случае необходимо определить большее по модулю число, производить вычитание чисел и присваивать разности знак большего по модулю числа. Поэтому для представления отрицательных чисел обычно применяют другие коды: обратный и дополнительный.

Обратный код

Чтобы представить двоичное отрицательное число в обратном коде необходимо поставить в знаковый разряд 1, а все остальные разряды числа изменить на противоположные.

Например, число +12 в 8-битном представлении записывается в прямом коде как b00001100, а число -12 в обратном коде - как b11110011. В обоих случаях старший бит числа является знаковым.

Обратный код $[A]_{об}$, если рассматривать его как число, является дополнением модуля исходного числа A до наибольшего числа без знака, помещающегося в разрядную сетку. В случае n - разрядной сетки имеем:

$$[A]_{об} = 2^n - 1 - |A|, \text{ если } A \text{ – целое число,}$$

$$[A]_{об} = 2 - 2^{-(n-1)} - |A|, \text{ если } A \text{ – двоичная дробь.}$$

Разрядная сетка длиной n обеспечивает запись в обратном коде отрицательных целых чисел от $b11\dots0 = -1$ до $b10\dots0 = -(2^{n-1} - 1)$. При 8-разрядной сетке это диапазон от -0 до -127. Для дробных чисел возможно представление от $b11\dots10 = -2^{-(n-1)}$ до $b100\dots00 = -(1 - 2^{n-1})$. При 8-разрядной сетке это числа от $-1/128$ до $-127/128$.

Обратный код имеет две различные записи нуля, как положительное и как отрицательное число. Например, в 8-разрядном представлении $+0 = b00000000$, а $-0 = b11111111$.

Изменение знака отрицательного числа, представленного в обратном коде, соответствует инвертированию его кода. В результате получается прямой код соответствующего положительного числа.

Дополнительный код

Дополнительный код $[A]_{доп}$ отрицательного числа A определяется выражением:

$$[A]_{доп} = 2^n - |A|, \text{ если } A \text{ – целое число,}$$

$$[A]_{доп} = 2 - |A|, \text{ если } A \text{ – двоичная дробь.}$$

Дополнительный код больше обратного на единицу: $[A]_{доп} = [A]_{об} + 1$. Поэтому для представления числа в дополнительном коде его можно сначала перевести в код обратный, а затем прибавить 1.

В области целых чисел дополнительный код позволяет представить числа от $b100\dots000 = -2^{n-1}$ до $b11\dots1111 = -1$. Восьмиразрядная сетка позволяет записать числа от 0 до -128.

В области дробных чисел можно представить числа от $b100\dots00 = -1$ до $b11\dots11 = 2^{-(n-1)}$. Восьмиразрядная сетка позволяет записать дробные числа от -1 до $-1/128$.

«Отрицательный» 0 в дополнительном коде отсутствует. При представлении положительных чисел прямым кодом, а отрицательных – дополнительным как положительный, так и отрицательный 0 изображаются одинаково: $-0 = +0 = b000\dots00$.

Изменение знака отрицательного числа, представленного в дополнительном коде, соответствует инвертированию его кода и добавлению 1 младшего разряда. В результате получается прямой код соответствующего положительного числа.

В табл. 1.5 приведены примеры восьмиразрядных положительных и отрицательных целых чисел от в диапазоне от +127 до -128 в прямом, обратном и дополнительном кодах.

Таблица 1.5.

Примеры записи целых восьмиразрядные чисел

Десятичное число	Прямой код	Обратный код	Дополнительный код
+127	b01111111		
+126	b01111110		
+7	b00000111		
+4	b00000100		
+3	b00000011		
+2	b00000010		
+1	b00000001		
+0	b00000000		
- 128			b1000 0000
- 127	b11111111	b1000 0000	b1000 0001
- 126	b11111110	b1000 0001	b1000 0010
- 7	b10000111	b1111 1000	b1111 1001
- 4	b10000100	b1111 1011	b1111 1100
- 3	b10000011	b1111 1100	b1111 1101
- 2	b10000010	b1111 1101	b1111 1110
- 1	b10000001	b1111 1110	b1111 1111
- 0	b10000000	b1111 1111	b0000 0000

1.6. Двоично-десятичные числа

Двоично-десятичное кодирование или BCD-код (*Binary Code Decimal*) используется для записи десятичных чисел с помощью двоичных символов.

В BCD-коде десятичного числа каждая десятичная цифра изображается четырьмя двоичными разрядами (тетрадой):

$$B = b_{n-1}b_{n-2}...b_1b_0 = \{a_3^{n-1}a_2^{n-1}a_1^{n-1}a_0^{n-1}\}_{n-1}\{a_3^{n-2}a_2^{n-2}a_1^{n-2}a_0^{n-2}\}_{n-2}... \{a_3^0a_2^0a_1^0a_0^0\}_0,$$

где: b_j - десятичные цифры,

a_i^j - двоичные разряды тетрады,

$i = 0..3$ - номер разряда внутри тетрады,

$j = 0..n-1$ - номер тетрады.

Разряды тетрад имеют определенный вес. Значение десятичной цифры b_j соответствует выражению:

$$b_j = a_3^j d_4 + a_2^j d_3 + a_1^j d_1 + a_0^j d_0,$$

где d_i - вес i -го разряда тетрады.

Наиболее распространенным BCD-кодом является код прямого замещения 8421 с весовыми коэффициентами, рассчитываемыми по формуле $d_j = 2^{i-1}$.

Наличие разрешенных и запрещенных комбинаций - важное свойство BCD-кодов. Из 16 кодовых комбинаций 6 запрещены. В коде 8421 запрещены комбинации: 1010, 1011, 1100, 1101, 1110, 1111. Это свойство отличает BCD – представление чисел от обычных позиционных систем счисления, в которых все комбинации - разрешенные. В тех случаях, когда в одной формуле приводятся двоичные и двоично-десятичные числа последние будем отмечать слева символами bcd. Например, $79 = b1001111 = bcd01111001$

В табл. 1.6 приведены примеры записи некоторых десятичных чисел в BCD-коде прямого замещения .

Таблица 1.6

Примеры записи десятичных чисел в BCD-коде 8421

Десятичные числа	BCD код 8421	Десятичные числа	BCD код 8421
10	0001 0000	20	0010 0000
11	0001 0001	21	0010 0001
12	0001 0010	22	0010 0010
13	0001 0011	33	0011 0011
14	0001 0100	34	0011 0100
15	0001 0101	45	0100 0101
16	0001 0110	56	0101 0110
17	0001 0111	67	0110 0111
18	0001 1000	78	0111 1000
19	0001 1001	89	1000 1001

Отрицательные двоично-десятичные числа могут быть представлены в прямом, обратном и дополнительном коде.

Если отрицательное число представлено тетрадами $b_i : B = - b_{n-1}b_{n-2}...b_1b_0$, то:

в прямом коде:

$$[B]_{пр} = 1b_{n-1}b_{n-2}...b_1b_0 ,$$

в обратном коде:

$$[B]_{об} = \bar{1}b_{n-1}\bar{b}_{n-2}...b_1\bar{b}_0 ,$$

в дополнительном коде:

$$[B]_{об} = \bar{1}b_{n-1}\bar{b}_{n-2}...b_1\bar{b}_0 ,$$

где: \bar{b} - дополнение тетрады до 9, $\bar{\bar{b}}$ - дополнение тетрады до 10.

Однако наличие в коде BCD-коде запрещенных комбинаций несколько видоизменяет представление отрицательных чисел. Инвертирование набора тетрад означает получение дополнения до $2^4 - 1 = 15$, а не до 9. Поэтому необходима коррекция полученного значения. Например, при переходе к отрицательным числам к каждой тетраде числа в BCD-коде добавляется 6 = b0110 и после этого производится инвертирование набора. Полученный результат представляет собой обратный код числа.

Например, запись двоично-десятичного числа – 256 в обратном и дополнительном коде выполняется в следующей последовательности:

$$\begin{array}{rcccc}
 256 & = & 0000 & 0010 & 0101 & 0110 \\
 + & & 0110 & 0110 & 0110 & 0110 & \text{коррекция} \\
 \hline
 & & 0110 & 1000 & 1011 & 1100 \\
 + & & 1001 & 0111 & 0100 & 0011 & \text{- 256 в обратном коде} \\
 \hline
 & & & & & 1 & \\
 + & & & & & & \\
 \hline
 & & 1001 & 0111 & 0100 & 0100 & \text{- 256 в дополнительном коде}
 \end{array}$$

1.7. Перевод чисел из одной системы счисления в другую

Перевод из десятичной системы

Для перевода целого десятичного числа в его двоичный (восьмеричный, шестнадцатеричный) эквивалент используется деление на 2 (8, 16), т.е. выполняется деление на основание новой системы счисления. В процессе такого деления последовательно, начиная с младшего разряда 2-го (8-го, 16-го) эквивалента, записывается остаток, если он получается на очередном этапе деления десятичного числа. В противном случае записывается ноль. Далее результат очередного деления опять делится на 2 (8, 16), если этот результат больше или равен 2 (8, 16). Если же результат меньше, то он прямо переписывается в старший разряд.

Например, перевод числа 53 в двоичную систему выполняется следующим образом:

$$\begin{array}{ccccccc}
 \underline{53:2} = & \underline{26:2} = & \underline{13:2} = & \underline{6:2} = & \underline{3:2} = & \underline{1} & \\
 \text{(младший бит)} & 1 & 0 & 1 & 0 & 1 & 1 \text{ (старший бит)}
 \end{array}$$

Ответ: 53 = b110101.

Аналогичным образом выглядит процесс перевода числа 128 в восьмеричную систему:

$$\begin{array}{cccc}
 \underline{128:8} = & \underline{16:8} = & \underline{2} & \\
 \text{(младший разряд)} & 0 & 0 & 2 \text{ (старший разряд)}
 \end{array}$$

Ответ: 128 = q200

Для дробных чисел требуется отдельная процедура перевода. Преобразование осуществляется умножением дроби на основание системы счисления, в которой дробь должна быть представлена. Если результат <1, то старшему значащему разряду присваивается значение 0; если больше 1, то присваивается 1. Результат формируется поразрядно, начиная со старшего разряда. Шаги описанной процедуры повторяются до тех пор, пока либо результат умножения не будет точно равен 1, либо не будет достигнута требуемая точность.

Процедуру преобразования десятичной дроби в двоичную рассмотрим на примере преобразования числа 0,375.

$$\begin{array}{cccc}
 0,375 \times 2 = & 0,75 \times 2 = & 1,5 - 1 = & 0,5 \times 2 = & 1 \\
 \text{(старший значащий бит)} & 0 & 1 & 1 & \text{(младший бит)}
 \end{array}$$

Ответ: 0,375 = b0,011.

Надо отметить, что не всегда путем повторения операций умножения можно достичь результата, точно равного 1. В таком случае процесс останавливается по достижению необходимой точности, а целую часть результата последней операции умножения присваивают младшему значащему разряду.

Например, перевод десятичного числа 0,3437510 в двоичное:

$$0,34375 \times 2 = 0,6875 \times 2 = 1,375 - 1 = 0,375 \times 2 = 0,75 \times 2 = 1,5 - 1 = 0,5 \times 2 = 1$$

(старший бит) 0 1 0 1 1 (младший бит)

Ответ: 0,3437510 = b0,01011

Для перевода десятичной правильной дроби в восьмеричную (шестнадцатеричную) надо умножить ее на 8 (16). Если очередное произведение правильная дробь, то, начиная со старшего разряда результата, записываются 0. Если произведение целое и меньше 8 (16), то оно прямо переписывается в соответствующий разряд результата.

Например, перевод числа 0,0625 в восьмеричную систему:

$$0,0625 \times 8 = 0,5 \times 8 = 4$$

(старший разряд) 0 4 (младший разряд)

Ответ: 0,0625 = q0,04

Перевод в десятичную систему

Перевод двоичного числа в десятичный его эквивалент можно выполнить при помощи формулы (1.2).

Например, перевод двоичного числа b110101 в десятичную систему выглядит следующим образом:

$$b110101 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 16 + 4 + 1 = 53.$$

А перевод числа q200 в десятичную систему представляется формулой:

$$q200 = 2 \cdot 8^2 + 0 \cdot 8^1 + 0 \cdot 8^0 = 128$$

Таким образом, при переводе числовой информации из одной позиционной системы счисления в другую все действия должны выполняться по правилам арифметики исходной системы счисления.

Перевод чисел из BCD-кода в двоичный

Пусть задано 4-х разрядное число в BCD-коде 8421: $V = b_3 b_2 b_1 b_0$, каждая десятичная цифра которого представлена в виде тетрады: $b_i = a_3^i a_2^i a_1^i a_0^i$. В данном случае $i = 0, 1, 2, 3$. Как известно, изображение числа V можно представить в следующем виде:

$$V = b_3 \cdot 10^3 + b_2 \cdot 10^2 + b_1 \cdot 10^1 + b_0 \cdot 10^0$$

Таким образом при переводе из кода BCD в двоичный код каждый i -тый разряд кода BCD надо умножить на 10^i . В то же время $10 = 8 + 2 = [A]_{доп}$, значит:

$$b_0 = \{a_3^0 a_2^0 a_1^0 a_0^0\} \times 1;$$

$$b_2 = \{a_3^2 a_2^2 a_1^2 a_0^2\} \times (2^3 + 2) \times (2^3 + 2) = \{a_3^2 a_2^2 a_1^2 a_0^2\} \times (2^6 + 2^5 + 2^2);$$

$$b_3 = \{a_3^3 a_2^3 a_1^3 a_0^3\} \times (2^3 + 2) \times (2^3 + 2) \times (2^3 + 2) = \{a_3^3 a_2^3 a_1^3 a_0^3\} \times (2^{10} + 2^7 + 2^5 + 2^4)$$

;

Умножение на 2^k означает сдвиг двоичного числа на k разрядов влево. Следовательно, перевод сводится к сдвигу соответствующих тетрад и их последующему суммированию. Нулевая тетрада не умножается. Первая тетрада сдвигается на 3 разряда влево и сохраняется, затем эта же тетрада сдвигается на 1 разряд влево, добавляется к сохраненной и опять сохраняется. Далее со второй тетрадой выполняются аналогичные процедуры и т. д. Все полученные результаты потетрадно складываются.

Упражнения

1. Записать в 8-разрядной сетке в прямом коде следующие двоичные числа: а) 0,0101010, б) $-0,1001001$, в) 0,1100110, г) -1000100 , д) $-0,1001100$.
2. Записать в дополнительном коде следующие числа: а) $-0,10101010$, б) $-0,10010010$, в) $-0,11001100$, г) $-0,10001000$, д) $-0,1001100$.
3. Записать в обратном коде следующие числа: а) $-0,10101010$, б) $-0,10010010$, в) $-0,11001100$, г) $-0,10001000$, д) $-0,1001100$.
4. Вычислить сумму $A+B$, представив отрицательные числа в дополнительном коде: а) $A=0,101010$, $B=-0,01001$ б) $A=-0,1001001$, $B=0,1110101$ в) $A=0,11001100$, $B=-0,01001$ г) $A=10001000$, д) $B=-10011001$.
5. Вычислить сумму $A+B$, представив отрицательные числа в обратном коде: а) $A=0,101010$, $B=-0,01001$ б) $A=-0,1001001$, $B=0,1110101$ в) $A=0,11001100$, $B=-0,01001$ г) $A=10001000$, д) $B=-10011001$.
6. Перевести числа A и B в двоично-десятичный код, сложить и перевести результат в десятичный код: а) $A=0,543$, $B=0,777$, б) $A=0,492$, в) $B=0,321$, г) $A=-109$, $B=375$ д) $A=-777$, $B=445$.
7. Перевести следующие числа из десятичной системы счисления в двоичную и из двоичной в шестнадцатеричную: а) 127, б) 36, в) 255, г) 155.
8. Перевести следующие числа из шестнадцатеричной системы в двоичную и из двоичной в десятичную: а) B9, б) 9A, в) 7F, г) C2, д) 56.
9. Перевести следующие числа из шестнадцатеричной системы непосредственно в десятичную: а) B2, б) 7A, в) 5E, г) CD, д) F6.
10. Перевести следующие числа из двоичной системы счисления в восьмеричную и из восьмеричной в десятичную с точностью 0,0001: а) 10101010, б) 10010010, в) 0,110011, г) 0,10001, д) 0,1001101.
11. Перевести следующие числа из десятичной системы в шестнадцатеричную без промежуточного перевода в двоичную с точностью до трех разрядов после запятой: а) 2002, б) 156, в) 0,255, г) ,934, д) 0,10017.
12. Перевести следующие числа из двоичной системы в десятичную: а) 10111010, б) 0,1001001, в) 0,11001100, г) 10001000, д) 0,1001100.
13. Перевести следующие числа из десятичной системы счисления в восьмеричную без промежуточного перевода в двоичную систему: а) 123, б) 0,456, в) 0,765, г) 1875, д) 0,555.
14. Перевести следующие числа в десятичную систему счисления: а) b1111111, б) hFFFF, в) q7777.
15. Перевести в двоично-десятичный код следующие числа: а) 67, б) 245, в) b11001100, г) b10001000, д) b01001100.
16. Перевести из двоично-десятичного кода в десятичный и двоичный следующие числа: а) bcd10101010, б) bcd10010010, в) bcd11001100, г) bcd10001000, д) bcd01001100.

2. АЛГЕБРА ЛОГИКИ

Алгебра логики, созданная в середине 18 века англичанином Дж. Булем (булева алгебра) оперирует с логическими переменными. Основополагающим законом алгебры логики является *закон исключения третьего*, согласно которому логические переменные, в отличие от переменных обычной алгебры, могут принимать только два значения. Переменные обычно обозначаются как и двоичные цифры символами 0 и 1. Операции над переменными записываются с помощью логических операций.

В электронных схемах операции выполняются с помощью логических элементов. При этом логические сигналы 0 и 1 задаются разными уровнями напряжения. Сигнал логического нуля обычно представляется низким уровнем напряжения VL (*Voltage Low*), логической единицы - высоким VH (*Voltage High*). Такая логика получила название *положительной*. В ряде случаев используют *отрицательную логику*, где логический ноль представляется высоким уровнем напряжения, а логическая единица - низким.

2.1. Логические функции

Функция от двоичных переменных $f(x_1, x_2, \dots, x_n)$ называется логической или булевой, если она также как и её аргументы, принимает только два значения: 0 и 1. Любая логическая функция может быть задана таблицей истинности, отражающей все возможные сочетания аргументов и соответствующие значения функции. В общем случае, для n входных переменных можно описать $N = 2^n$ различных логических функций: для одной переменной – 4, для двух – 16 и т.д. Логические функции одной и двух переменных имеют свои оригинальные названия и математические обозначения. Все они приведены в таблицах 2.1 и 2.2.

Таблица 2.1.

Логические функции одной переменной

Обозначения	Наименования	x	
		0	1
$f_1 = 0$	Константа 0	0	0
$f_2 = x$	Повторение	0	1
$f_3 = \bar{x}$	Инверсия	1	0
$f_4 = 1$	Константа 1	1	1

Таблица 2.2

Логические функции двух переменных

Обозначение	Наименование	x1x2			
		00	01	10	11
$f_1 = x_1 x_2$	Конъюнкция	0	0	0	1
$f_2 = x_1 \vee x_2$	Дизъюнкция	0	1	1	1
$f_3 = x_1 \rightarrow x_2$	Импликация x_1 в x_2	1	1	0	1
$f_4 = x_1 \leftarrow x_2$	Импликация x_2 в x_1	1	0	1	1
$f_5 = x_1 \overline{\rightarrow} x_2$	Запрет x_2	0	0	1	0
$f_6 = x_1 \overline{\leftarrow} x_2$	Запрет x_1	0	1	0	0
$f_7 = x_1 \sim x_2$	Эквивалентность	1	0	0	1
$f_8 = x_1 \oplus x_2$	Сумма по модулю 2	0	1	1	0
$f_9 = x_1 / x_2$	Штрих Шеффера	1	1	1	0

$f_{10} = x1 \downarrow x2$	Стрелка Пирса	1	0	0	0
$f_{11} = x1$	Повторение $x1$	0	0	1	1
$f_{12} = x2$	Повторение $x2$	0	1	0	1
$f_{13} = 1$	Константа 1	1	1	1	1
$f_{14} = 0$	Константа 0	0	0	0	0
$f_{15} = \overline{x1}$	Инверсия $x1$	1	1	0	0
$f_{16} = \overline{x2}$	Инверсия $x2$	1	0	1	0

Логические функции для произвольного количества входных переменных $f(x1, x2, \dots, xn)$ могут составляться из простейших функций одной и двух переменных. При этом не обязательно использовать все известные функции, приведенные в табл. 2.1 и 2.2, а можно ограничиться определенным набором, в который входит всего несколько функций. При этом полагается, что в выражении любую логическую функцию, включенную в набор, можно использовать неограниченное количество раз. Например, $f = x1x2 \vee \overline{x1}x3 \vee x2\overline{x3}$ - некоторая функция трех переменных выражается с помощью простейших операций конъюнкции, дизъюнкции и инверсии.

Набор логических функций считается *функционально-полным*, если он позволяет выразить любую логическую функцию. В алгебре логики доказывается, что функционально-полными являются наборы функций:

- $f_1 = x1x2$ (конъюнкция), $f_2 = x1 \vee x2$ (дизъюнкция) и $f_3 = \overline{x}$ (инверсия),
- $f_1 = x1x2$ (конъюнкция), $f_8 = x1 \oplus x2$ (сумма по модулю 2) и $f_3 = \overline{x}$ (инверсия),

Функциональной полнотой обладают также наборы из двух функций:

- $f_2 = x1 \vee x2$ (дизъюнкция) и $f_3 = \overline{x}$ (инверсия),
- $f_1 = x1x2$ (конъюнкция) и $\mathcal{U}=\mathcal{V}_f$ (инверсия),

и даже наборы, содержащие только одну функцию:

- $f_9 = x1 / x2$ (Штрих Шеффера),
- $f_{10} = x1 \downarrow x2$ (Стрелка Пирса).

Существуют и другие, менее распространенные наборы, логических функций.

Операции, входящие в функционально-полные наборы логических функций считаются основными и чаще всего используются при построении цифровых схем. Они, как правило, имеют несколько альтернативных названий и символических обозначений. К таким функциям относятся:

- $f_1 = x1x2 = x1 \wedge x2 = x1 \& x2$ (конъюнкция, логическое умножение, операция И, AND),
- $f_2 = x1 \vee x2$ (дизъюнкция, логическое сложение, операция ИЛИ, OR),
- $f_7 = x1 \sim x2$ (эквивалентность, инверсия исключающего ИЛИ, XNOR),
- $f_8 = x1 \oplus x2$ (сумма по модулю 2, исключающее ИЛИ, неэквивалентность, XOR).

- $f_9 = x1 / x2$ (Штрих Шеффера, операция И-НЕ, NAND),
- $f_{10} = x1 \downarrow x2$ (Стрелка Пирса, операция ИЛИ-НЕ, NOR).
- $f_{15} = \overline{x1} = !x1 = \neg x1$ (инверсия, операция НЕ, NOT),

2.2. Законы и теоремы алгебры логики

В булевой алгебре действуют свои законы и теоремы. Многие из них имеют привычный для обычной алгебры вид. Некоторые противоречат нашим представлениям.

Коммутативный (переместительный) закон записывается в привычном для нас виде, справедлив как для логического сложения, так и для логического умножения:

$$x1 \cdot x2 = x2 \cdot x1 \qquad x1 \vee x2 = x2 \vee x1 \qquad (2.1)$$

Ассоциативный (сочетательный) закон также действует в обычной арифметике и справедлив для логического сложения и логического умножения:

$$x1 \cdot (x2 \cdot x3) = (x1 \cdot x2) \cdot x3 \qquad x1 \vee (x2 \vee x3) = (x1 \vee x2) \vee x3 \qquad (2.2)$$

Дистрибутивный (распределительный) закон. Только первая запись этого закона не вызывает у нас удивления. Вторая явно противоречит представлениям обычной алгебры.

$$x1(x2 \vee x3) = x1x2 \vee x1x3 \qquad x1 \vee (x2x3) = (x1 \vee x2) \cdot (x1 \vee x3) \qquad (2.3)$$

Все остальные правила и теоремы относятся только к алгебре логики. В обычной алгебре они явно бессмысленны.

Правило склеивания:

$$x1(x1 \vee x2) = x1 \qquad x1 \vee x1x2 = x1 \qquad (2.4)$$

Правило повторения:

$$x \cdot x = x \qquad x \vee x = x \qquad (2.5)$$

Правило отрицания:

$$x \cdot \bar{x} = 0 \qquad x \vee \bar{x} = 1 \qquad (2.6)$$

Правило двойного отрицания:

$$\overline{(\bar{x})} = x \qquad (2.7)$$

Теорема де Моргана

$$\overline{(x1 \cdot x2)} = \bar{x1} \vee \bar{x2} \qquad \overline{(x1 \vee x2)} = \bar{x1} \cdot \bar{x2} \qquad (2.8)$$

Последняя теорема (2.8) имеет очень широкое применение. Она позволяет перейти от логического умножения к логическому сложению (и обратно) и, в принципе, позволяет строить различные логические схемы, используя ограниченный набор логических элементов.

2.3. Описание логических функций

Таблица истинности

Логическая функция нескольких переменных однозначно задается в виде *таблицы истинности*, в которой для каждому набору входных переменных ставится в соответствие определенное значение функции. В общем случае, для n переменных возможно $N = 2^n$ различных сочетаний и, следовательно, таблица истинности для функции n переменных должна иметь $N = 2^n$ строк. В качестве примера, в таблице 2.3 описана логическая функция $y = x2 \oplus x1 \oplus x0$.

Таблица 2.3 .

Пример таблицы истинности функции трех переменных

x_2	x_1	x_0	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Совершенная дизъюнктивная нормальная форма

Совершенная дизъюнктивная нормальная форма (СДНФ) является наиболее распространенной формой аналитического представления логической функции.

Для представления функции в виде СДНФ каждому набору переменных в таблице истинности ставится в соответствие *минтерм* – конъюнкция всех переменных, которые входят в выражение в прямом виде, если значение данной переменной в наборе равно 1, либо в инверсном виде, если значение переменной равно 0. Для n входных переменных можно составить $N = 2^n$ минтермов. Например, все минтермы функции трех переменных приведены в таблице 2.4

Таблица 2.4

Минтермы функции трех переменных

Набор переменных			Номер набора	Минтерм
x_2	x_1	x_0		
0	0	0	0	$m_0 = \bar{x}_2 \bar{x}_1 \bar{x}_0$
0	0	1	1	$m_1 = \bar{x}_2 \bar{x}_1 x_0$
0	1	0	2	$m_2 = \bar{x}_2 x_1 \bar{x}_0$
0	1	1	3	$m_3 = \bar{x}_2 x_1 x_0$
1	0	0	4	$m_4 = x_2 \bar{x}_1 \bar{x}_0$
1	0	1	5	$m_5 = x_2 \bar{x}_1 x_0$
1	1	0	6	$m_6 = x_2 x_1 \bar{x}_0$
1	1	1	7	$m_7 = x_2 x_1 x_0$

Логическая функция в СДНФ представляется в виде дизъюнкции:

$$y = m_0 \cdot y_0 \vee m_1 y_1 \vee \dots \vee m_{q-1} y_{q-1}, \quad (2.9)$$

где y_i и m_j – значения логической функции и минтерм, соответствующие i -му набору переменных ($i = 0 \dots q-1$).

- Таким образом, при переходе от таблицы истинности к СДНФ необходимо:
- В таблице истинности выделить строки, в которых функция принимает единичные значения;
 - Для каждой выделенной строки составить минтерм.
 - Записать логическую сумму всех составленных минтермов.

Например, функция, заданная таблицей истинности 2.3 принимает единичные значения на четырех наборах входных переменных и в совершенной дизъюнктивной нормальной форме записывается в виде дизъюнкции четырех минтермов:

$$y = \bar{x}_2 \bar{x}_1 x_0 \vee \bar{x}_2 x_1 \bar{x}_0 \vee x_2 \bar{x}_1 \bar{x}_0 \vee x_2 x_1 x_0$$

Карта Карно

Компактной и очень удобной формой записи логической функции, используемой наряду с таблицей истинности, является карта Карно. Карта Карно функции n переменных представляет собой прямоугольник, разделенный на клетки, каждой из которых соответствует один из $N = 2^n$ минтермов. Расположение минтермов определяется номером столбца и номером строки, на пересечении которых находится клетка. Комбинации входных переменных распределяются по двум сторонам прямоугольника, а значения функции записываются в клетки таблицы. Клетки карты Карно считаются соседними, если их минтермы различаются состоянием только одной переменной.

Карта Карно для функции двух переменных содержит четыре клетки и имеет форму квадрата. На рис. 2.1а в клетки карты записаны соответствующие минтермы, а на рис. 2.1б, в качестве примера, в карту записана рассмотренная ранее булева функция ЭКВИВАЛЕНТНОСТЬ $f_7 = x_1 \sim x_0$ из таблицы 2.2.

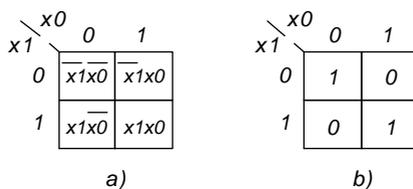


Рис. 2.1 . Карта Карно для функции двух переменных

В карте Карно двух переменных (рис. 2.1) каждая клетка граничит с двумя соседними.

Карта Карно для функции трех переменных состоит из 8 клеток и имеет обычно 2 строки и четыре столбца (рис. 2.2). На верхней стороне прямоугольника каждому столбцу ставится в соответствие одна комбинация входных переменных x_0 и x_1 . Причем, при переходе от каждого столбца к соседнему имеет право измениться только одна переменная, а первый и последний столбцы карты также считаются соседними. Для примера в карту занесена функция $y = x_2 \oplus x_1 \oplus x_0$ (табл. 2.3).

x_1x_0	00	01	11	10
x_2	0	1	0	1
1	1	0	1	0

Рис. 2.2. Карта Карно для функции трех переменных

В карте трех переменных (рис. 2.2) каждая клетка имеет три соседние. Например, клетка с минтермом $\overline{x_2x_1x_0}$ имеет три соседние клетки с минтермами: $x_2x_1x_0$, $x_2x_1\overline{x_0}$ и $\overline{x_2x_1x_0}$.

В карте Карно для функции четырех переменных 16 клеток размещены в четырех столбцах и четырех строках. Две переменные x_0 и x_1 располагаются наверху квадрата, а две другие x_2 и x_3 - слева (рис. 2.3).

x_1x_0	00	01	11	10
x_2x_3	00	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

Рис. 2.3. Карта Карно для функции четырех переменных

В отличие от предыдущего случая здесь каждой строчке таблицы соответствует определенная комбинация двух переменных x_2 и x_3 . При переходе от каждой строки к соседней меняется только одна переменная, а первая и последняя строки карты также как и крайние столбцы считаются соседними. Каждая клетка карты имеет четыре соседние клетки. Для примера в карту занесена логическая функция: $y = x_0 \oplus x_1 \oplus x_2 \oplus x_3$.

При числе переменных равном или большем пяти, отобразить графически функцию в виде одной плоской карты Карно невозможно. В таких случаях можно строить комбинированную карту, состоящую из совокупности более простых карт.

Минимизация логических функций

Законы и теоремы булевой алгебры позволяют минимизировать (упростить) логическое выражение, представленное в совершенной дизъюнктивной нормальной форме. При небольшом количестве переменных минимизацию удобно осуществлять непосредственно по карте Карно. Если в карте Карно встречаются группы из 2-х, 4-х, 8-ми и т.д. соседних ячеек, содержащих единицы, которые можно выделить контуром в виде квадрата или прямоугольника, то такая группа может быть описана одним логическим произведением. В это произведение входят только неизменные для всех ячеек данной группы переменные. Например, в карте Карно четырех переменных рис. 2.14 можно выделить группу из четырех клеток в первом столбце, группу из четырех соседних клеток в верхнем правом углу и группу из двух соседних клеток в нижней строке.

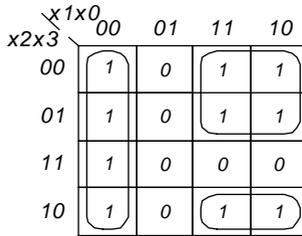


Рис. 2.14. Минимизация функции четырех переменных

В результате минимизированная функция представляет собой сумму трех произведений, соответствующих отдельным группам: $y = \bar{x}1\bar{x}0 \vee \bar{x}2x1 \vee \bar{x}3x2x1$.

3. ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ

Условные обозначения

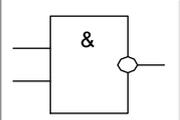
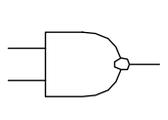
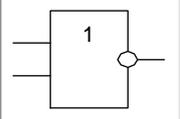
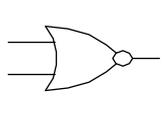
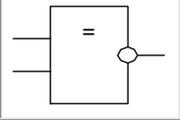
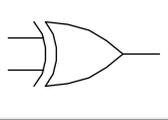
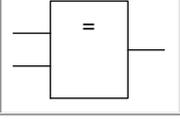
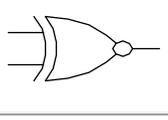
Логические элементы являются специализированными цифровыми схемами, предназначенными для аппаратной реализации логических функций. Условное обозначение такого элемента не зависит от технологии его изготовления.

В настоящее время в мире существует и развивается несколько систем условных обозначений. В нашей стране обозначения логических элементов в схемах выполняются по стандарту ГОСТ 2.743-82. В программных продуктах наиболее распространена американская система milspec. Условные графические обозначения наиболее распространенных элементов в отечественной и американской системе приведены в таблице 3.1.

Таблица 3.1

Условные обозначения логических элементов

Логическая функция элемента	Отечественное обозначение	Американское обозначение
НЕ (Инверсия, NOT)		
И (Конъюнкция, AND)		
ИЛИ (Дизъюнкция, OR)		

И-НЕ (Штрих Шеффера, NAND)		
ИЛИ-НЕ (Стрелка Пирса, NOR)		
ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR)		
Инверсия ИСКЛЮЧАЮЩЕГО ИЛИ (XNOR)		

Основные параметры

Логические элементы характеризуются определенным набором параметров, приводимых в справочной документации. Большинство параметров имеют четкое однозначное определение, что обеспечивает возможность сравнения логических элементов различных типов.

Статические параметры

Уровни напряжений и пороги переключений

Основной статической характеристикой логического элемента является *передаточная характеристика* - зависимость выходного напряжения VO (*Voltage Output*) от напряжения на входе VI (*Voltage Input*). На рис. 3.1, в качестве примера, показана типовая передаточная характеристика логического элемента НЕ (NOT).

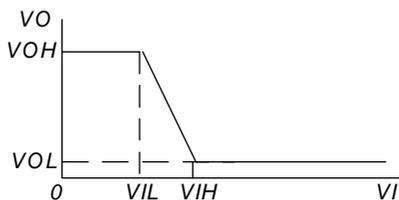


Рис. 3.1. Передаточная характеристики логического элемента НЕ (NOT)

Характеристика имеет три четко выраженных участка. Первый соответствует выходному напряжению логического нуля VOL (*Voltage Output Low*), второй - выходному напряжению логической единицы: VOH (*Voltage Output High*), третий - переходу из нуля в

единицу. Входные напряжения, определяющие границы участков, называются *порогами переключения*: V_{IL} (*Voltage Input Low*) и V_{IH} (*Voltage Input High*), а разность напряжений логической единицы и логического нуля - *логическим перепадом*:

$$\Delta V = V_{OH} - V_{OL} . \quad (3.1)$$

При построении схемы сигналы с выходов одних элементов подаются на входы других. Сигналы могут искажаться *помехами*, изменяющими их по величине. Для сигнала низкого уровня нежелательна помеха положительной полярности, повышающая уровень напряжения. *Запас помехоустойчивости по уровню 0* определяется разностью низшего порога переключения и уровня логического нуля: $V_{IL} - V_{OL}$.

Для сигнала высокого уровня, наоборот, опасна помеха отрицательной полярности, снижающая его уровень. *Запас помехоустойчивости по уровню 1* определяется разностью выходного напряжения логической единицы и высшим порогом переключения: $V_{OH} - V_{IH}$:

Для повышения помехоустойчивости необходимо увеличивать логический перепад ΔV и уменьшать ширину области переключения $V_{IH} - V_{IL}$ (рис. 3.1). Идеальная передаточная характеристика, обеспечивающая максимальный запас помехоустойчивости, равный $V_{CC}/2$, где V_{CC} - напряжение источника питания логического элемента, соответствует условиям: $V_{OL}=0$; $V_{OH}=V_{CC}$; $V_{IL}=V_{IH}=V_{CC}/2$. На рис 3.2 показана такая идеальная характеристика, получить на практике, которую никогда не удастся.

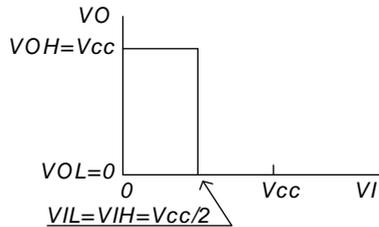


Рис. 3.2. Идеальная передаточная характеристика логического элемента

Эффективным способом повышения помехоустойчивости схем является создание передаточной характеристики с прямоугольной петлей гистерезиса (рис. 3.3).

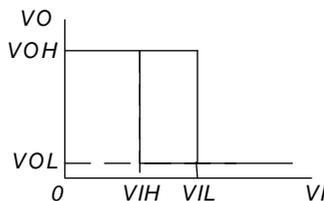


Рис. 3.3.. Передаточная характеристика с петлей гистерезиса

В этом случае: $V_{IL} > V_{IH}$ и в пределе (при $V_{IL}=V_{CC}$ и $V_{IH}=0$) запас помехоустойчивости достигает напряжения питания V_{CC} .

Входные токи

Входные токи логического элемента могут быть различны при различных значениях входного напряжения. Входной ток логического нуля I_{IL} обычно для логического элемента является вытекающим током, а входной ток логической единицы I_{IH} - втекающим.

Нагрузочная способность

Нагрузочная способность логического элемента определяется значением *коэффициента разветвления по выходу*, определяющего максимальное число входов аналогичных элементов, которые можно одновременно подключать к его выходу. Обычно значение коэффициента разветвления находится в пределах от 4 до 10. Мощные схемы допускают подключение до 30÷50 входов.

Потребляемая мощность

Потребляемая мощность логического элемента зависит от его состояния. Потребляемые от источника питания V_{CC} токи: I_{CC0} при $VO=VOL$ и I_{CC1} при $VO=VOH$ обычно значительно различаются. Поэтому для характеристики элемента используется средняя потребляемая мощность в статическом режиме, определяемая по формуле:

$$P = 0,5 \cdot V_{CC}(I_{CC0} + I_{CC1}) \quad (3.2)$$

Для уменьшения потребляемой мощности необходимо снижать напряжение питания V_{CC} , однако это однозначно приводит к ухудшению помехоустойчивости элемента.

Динамические параметры

Задержки распространения сигнала

Задержки распространения сигнала, характеризующие быстродействие логического элемента, определяются по *переходным характеристикам* элементов (рис. 3.4). Переходная характеристика элемента, отражающая изменения выходных VO и входных VI сигналов элемента во времени, снимается в цепи последовательно включенных схем.

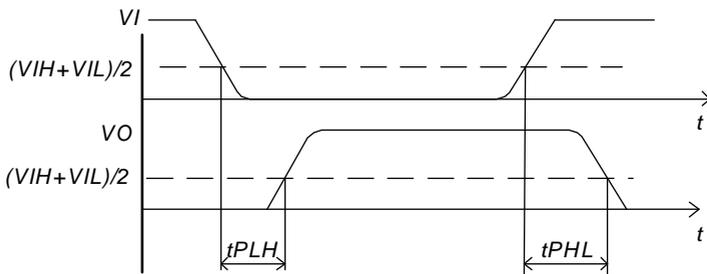


Рис. 3.4. Переходные характеристики логического элемента HE

Задержки распространения t_{PHL} (*Propagation Delay time High to Low Level*) и t_{PLH} (*Propagation Delay time Low to High Level*) определяются по переходным характеристикам как промежутки времени между моментами пересечения входными и выходными сигналами среднего порога переключения $(V_{IH}+V_{IL})/2$.

Работа переключения

Важным динамическим параметром логического элемента является также *работа переключения*, определяющая затраты энергии на перенос одного бита информации:

$$A_S = P \cdot tPD, \quad (3.3)$$

где $tPD = 0,5(tPHL+tPLH)$ - среднее время задержки переключения (*Propagation Delay*).

В настоящее время величина A_S достигла 10^{-4} пДж и продолжает снижаться. Теоретически минимальная работа переключения равна энергии элементарного шумового выброса: $\mathcal{E} = kT$, где k - постоянная Больцмана, T - абсолютная температура. При комнатной температуре $T \approx 300^\circ K$: $\mathcal{E} \approx 4 \cdot 10^{-9}$ пДж.

Практически все рассмотренные параметры логических элементов зависят от напряжения питания. Снижение напряжения уменьшает потребляемую мощность, ухудшает помехоустойчивость, снижает быстродействие и нагрузочную способность логических элементов. Поэтому наряду с номинальным напряжением для всех микросхем указывается его допустимые отклонения. Обычно они составляют 5÷10% номинального значения.

Температура также существенно влияет абсолютно на все параметры логических элементов. Все элементы имеют рабочий диапазон температур, в котором гарантируется соответствие их параметров указанным значениям. Для микросхем общего применения этот диапазон составляет $-10 \div +70$ °С. Для микросхем, предназначенных для работы в сложных условиях (бортовая аппаратура, полевые условия и др.), рабочий диапазон температур расширен до $-60 \div +125$ °С.

Классификация логических элементов

В процессе развития микроэлектроники схемы логических элементов претерпевали значительные изменения. Требования повышения быстродействия, помехоустойчивости, степени интеграции микросхем, снижения потребляемой мощности оказались очень противоречивыми. Единая технология, удовлетворяющая одновременно всем перечисленным критериям, до настоящего времени не найдена. Развитие идет одновременно в нескольких направлениях.

По напряжению питания все выпускаемые в мире логические элементы делятся на группы с напряжениями питания: 5 вольт, 3,3 вольта, 2,5 вольта и 1,8 вольта.

По технологии изготовления логические элементы делятся на 3 группы:

- биполярная логика (*bipolar logic*).
- комплементарная МОП-логика (*CMOS logic*).
- совмещенная логика (*BiCMOS logic*).

На рис. 3.5 приведена динамика развития стандартной логики, показывающая практически все производимые в мире семейства микросхем. Некоторые из них находятся на стадии появления, другие на стадиях развития и наибольшей популярности, ряд семейств уже снимается с производства и не рекомендуется к использованию в новых разработках. Каждое из семейств имеет периоды подъема, наибольшей популярности и спада.

переход “база-эмиттер” транзистора $VT1$ к источнику сигнала. Величина базового тока транзистора задается резистором $R1$ и практически не зависит от числа входов с нулевым потенциалом. Потенциал базы транзистора $VT1$ в этом случае равен: $VOL+0,7$, где $0,7 В$ - падение напряжения на открытом переходе “база-эмиттер”. Переход “база-коллектор” транзистора $VT1$, включенный последовательно переходу “база-эмиттер” транзистора $VT2$, в этом случае открыться не может, так как $VOL < 0,7 В$. Базовый ток транзистора $VT2$ равен нулю и транзистор находится в режиме отсечки. *Двухтактный* выходной каскад логического элемента состоит из транзистора $VT4$ и эмиттерного повторителя на транзисторе $VT3$. При запертом транзисторе $VT2$ ток от источника питания V_{cc} через резистор $R2$ поступает в базу транзистора $VT3$. Через открытый транзистор $VT3$, диод $VD3$ и резистор $R4$ выход логического элемента соединяется с источником питания. Выходное напряжение высокого уровня в этом режиме $VOH \approx 2,6 В$.

Если на всех входах логического элемента присутствуют сигналы высокого уровня, то ток базы транзистора $VT1$ через открытый переход “база-коллектор” поступает в цепь базы транзистора $VT2$. Транзистор $VT2$ переходит в открытое состояние. Часть его эмиттерного тока поступает в базу транзистора $VT4$. В результате транзистор переходит в состояние насыщения, а выходное напряжение логического элемента будет близко к нулю. Уровень логического нуля VOL определяется напряжением насыщения транзистора, не превышающем $0,3 В$.

Таким образом, выходной сигнал $Y=0$ появляется только при условии $X1=X2=1$, что соответствует логической функции И-НЕ. Как известно, логический элемент И-НЕ представляет собой функционально полную систему и, являясь базовым элементом ТТЛ-логики, обеспечивает построение на своей основе множества сложных логических схем.

В схеме логического элемента И-НЕ (рис. 3.6) на входах элемента установлены диоды $VD1$ и $VD2$, защищающие входы от пробоя при появлении на них импульсов отрицательной полярности. Возникновение таких импульсов возможно из-за накопления зарядов в емкости проводников печатных плат.

Резистор $R4$ предназначен для ограничения сквозного тока через транзисторы $VT4$ и $VT5$ в моменты переключения схемы, когда один из транзисторов открывается, а второй закрывается. Импульсы сквозного тока при переключениях элементов являются источником больших импульсных помех в цепях питания микросхем. Для уменьшения помех в цепи питания рекомендуется устанавливать керамические конденсаторы емкостью $0,1$ мкФ.

Более поздние разработки элементов биполярной логики имеют в схеме транзисторы с диодами Шоттки. К ним относятся семейства S (*Schottky*) и LS (*Low Schottky*). Последнее семейство характеризуется пониженным энергопотреблением.

Несколько измененные схемы логических элементов имеют так называемые *перспективные* семейства биполярной логики. Среди существующих трех разновидностей перспективных семейств выпущенное фирмой *Fairchild Camera and Instrument Corporation* семейство F (*Fairchild*) считается базовым. Семейство AS (*Advanced Schottky*) - быстродействующее, а семейство ALS (*Advanced Low Schottky*) - экономичное.

В таблице 3.2. приведены сравнительные параметры логических элементов различных семейств биполярной логики.

Таблица 3.2

Сравнительные данные элементов биполярной логики

Семейство	Отечественный аналог	P (мВт)	tPD (нс)	A_s (нДж)	VOL (В)	VOH (В)
-----------	----------------------	-----------	------------	-------------	-----------	-----------

Семейство	Отечественный аналог	P (мВт)	tPD (нс)	A_s (пДж)	VOL (В)	VOH (В)
TTL	K155	10	9	90	2,4	0,4
S	K531	19	3	57	2,6	0,45
LS	K555	2	9,5	19	2,6	0,45
F	K1531	4	3	12	2,5	0,45
ALS	K1533	1,2	4	4,8	2,5	0,4
AS	-	8	1,75	14	2,4	0,45

Несмотря на то, что семейства биполярной логики имеют несколько различные уровни сигналов, их можно использовать совместно. Коэффициенты разветвления элементов биполярной логики обычно равны 10. Элементы с *повышенной нагрузочной способностью* могут иметь коэффициент разветвления в пределах 30÷50.

CMOS- логика

Комплементарная логика CMOS (*Complimentary Metal Oxide Semiconductor*) - наиболее современный тип логики. Основным элементом логики - комплементарная пара транзисторов, состоящая из двух MOS-транзисторов с каналами n- и p- типа (рис. 3.6)

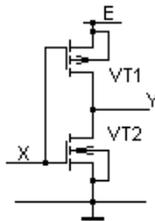


Рис. 3.7. Инвертор CMOS-логики

При нулевом входном сигнале $V_I < V_{IL}$ транзистор VT_2 открыт, а транзистор VT_1 - закрыт. В результате на выходе схемы $V_O = V_{OH} \approx V_{CC}$. В противоположном случае, при $V_I > V_{IH}$, открытым является транзистор VT_1 , а закрытым - VT_2 . На выходе элемента: $V_O = V_{OL} \approx 0$. Таким образом, комплементарная пара транзисторов является инвертором, реализующим функцию $Y = \overline{X}$. Особенностью такого инвертора является высокое, практически равное напряжению питания, значение логической единицы, что обуславливает очень высокую помехоустойчивость схемы. Напряжение питания элемента CMOS-логики может быть любым в диапазоне от 3 до 15 В. С увеличением напряжения питания увеличивается запас помехоустойчивости схемы.

Потребляемый схемой от источника питания ток равен нулю в любом из рассмотренных статических состояний элемента. В импульсном режиме энергия источника питания расходуется на перезаряд емкости нагрузки элемента C . Она зависит от напряжения питания схемы V_{CC} и частоты переключения f .

$$P = C \cdot f \cdot V_{CC}^2 . \quad (3.4)$$

Для современных схем потребляемая логическим элементом мощность составляет примерно 0,1÷0,3 мкВт/кГц. Энергия на одно переключение элемента (менее 0,05 пДж/кГц) значительно меньше, чем у любых других серий логических элементов.

3.8. Для реализации функции И-НЕ используется схема на четырех транзисторах рис.

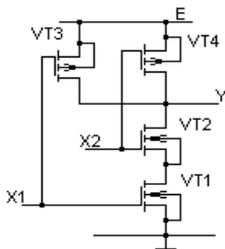


Рис. 3.8. Логический элемент И-НЕ CMOS-логики

В этой схеме сигнал логического нуля на выходе формируется только при одновременном открытии двух включенных последовательно транзисторов типа p-n-p: и VT1 VT2, т.е. при условии $X1=X2=1$.

При изменении последовательного соединения транзисторов на параллельное получается схема логического элемента ИЛИ-НЕ (рис. 3.9).

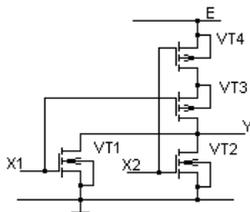


Рис. 3.9. Логический элемент ИЛИ-НЕ CMOS-логики

Первые микросхемы CMOS семейства CD4000A созданы фирмой RCA в начале 70-х годов (отечественный аналог - серия 176). Эти схемы имели напряжение питания +9В. Поскольку быстродействие CMOS-микросхем пропорционально напряжению питания, то в последующем напряжение питания элементов повышается до 15 В. Усовершенствованная серия CD4000B (отечественный аналог 561) при напряжении питания 15В обеспечивает время задержки 50 нс, а более поздняя серия LOGMOS 4000 фирмы Philips при том же напряжении питания 15В - задержку 15 нс. Быстродействующие CMOS-семейства конкурирует по функциям и конфигурации с перспективными биполярными семействами. Большой выбор современных CMOS серий предлагает фирма *Texas Instruments*.

Микросхемы семейств AC/ACT (*Advanced CMOS Logic*) производятся с использованием 1-микронной CMOS технологии. AC устройства имеют CMOS-совместимые входы, а ACT входы совместимые с TTL схемами.

Микросхемы АНС/АНСТ (*Advanced High-Speed CMOS Logic*) имеют высокое быстродействие при малом потреблении и низком уровне шумов. АНС схемы имеют CMOS-совместимые входы, а АНСТ - входы совместимые с TTL схемами.

ALVC (*Advanced Low-Voltage CMOS Technology*) является одной из наиболее производительных серий с 3 вольтовым питанием. Производится по 0,6 микронной CMOS технологии.

Сравнительные данные CMOS-микросхем

Семейство	Отечественный аналог	Vcc (В)	tPD (нс)
CD4000A	176	9	
CD4000B	561	15	25
HC	1564	4.5	8.8
HCT		4.5	8
AC	1554	5.5	4.5
ACT		5	8
AHC		2..5,5	5
AHCT			
LVC			
ALVC		3.3	2.2

BiCMOS логика

Микросхемы совмещенной логики BiCMOS (*Bipolar CMOS*) появились в начале 80-х годов. Они удачно совместили в себе высокое быстродействие TTL-логики и низкое энергопотребление микросхем CMOS.

На рис. 3.10 изображен инвертор BiCMOS-логики, соединяющий в себе логическую схему микросхем CMOS и выходной каскад TTL.

Рис. 3.10. Инвертор BiCMOS-логики

В статическом состоянии этот элемент работает как обычный CMOS-инвертор на транзисторах VT1 и VT2, у которого последовательно с каналом каждого транзистора включены резисторы R1 и R2, сопротивления которых сравнимы с сопротивлениями каналов в открытом состоянии. Статические токи в CMOS-инверторе отсутствуют, отсутствуют напряжения на резисторах R1 и R2 и биполярные транзисторы закрыты. При переключениях происходит перезаряд емкости нагрузки логического элемента Cн через открывающиеся транзисторы VT1 и VT2. Протекающие при этом через резисторы R1, R2 токи, открывают биполярные транзисторы VT3 и VT4, тем самым существенно ускоряя перезаряд емкости. Таким образом, использование биполярных транзисторов позволяет нейтрализовать влияние емкости нагрузки, являющейся одним из основных факторов, ограничивающих быстродействие CMOS-микросхем.

ABT (*Advanced BiCMOS Technology*) семейство микросхем второго поколения в основном предназначенных для шинных интерфейсов. Эти изделия производятся с использованием 0,8-микронной технологии, что обеспечивает выходной ток высокого уровня до 64 мА и задержку распространения ниже 5 наносекунд, при сохранении очень малой потребляемой мощности. Отдельные микросхемы этой серии обеспечивают чрезвычайно высокий выходной ток (до 180 мА).

LVT (*Low-Voltage BiCMOS Technology*) серия разработана по 0,8-микронной, специально для работы с пониженным 3-вольтовым питанием. Основное функциональное назначение микросхем LVT серии – работа с шинами данных. Так же как 5-вольтовая ABT серия LVT имеет выходные токи до 64 мА и характерную задержку распространения не

более 4 наносекунд для шинных формирователей, при этом тактовая частота может превышать 150 МГц.

Входы микросхем BiCMOS часто снабжаются встроенными подтягивающими резисторами (*pull-up resistors*) или запоминающими (*Bus Hold*) цепями.

Подтягивающий резистор соединяет вход микросхемы с источником питания и обеспечивает появление сигнала высокого уровня на неподключенном (свободном) входе. Базовое значение такого входа считается равным 1.

Bus Hold цепь представляет собой пару инверторов, вводящих небольшую положительную обратную связь (рис. 3.10).

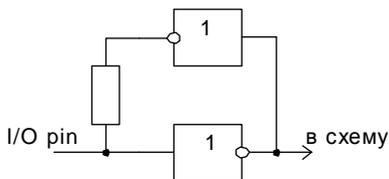


Рис. 3.10. *Bus Hold* цепь

Благодаря такой «запоминающей» цепи на свободном входе сохраняется напряжение, соответствующее сигналу в момент времени, предшествующий отключению. В этом случае отпадает необходимость установки на входах подтягивающих резисторов.

Для увеличения помехоустойчивости при работе на шину в микросхемы вводят встроенные согласующие демпфирующие резисторы (*Series-damping resistors*) сопротивлением порядка 25 Ом. Они позволяют согласовать работу микросхем с линиями, имеющими волновое сопротивление 25 Ом.

В таблице 3.2 приведены некоторые сравнительные данные BiCMOS микросхем.

Таблица 3.2

Сравнительные данные BiCMOS микросхем

СЕРИЯ	Питание (В)	Задержка (нс)	Частота (МГц)	IOH, мА	IOL, мА
74ABT	5	3.6	200	-32	64
74LVT	3.3	2.2	>150	-32	64

Сопряжение микросхем

Вопросы сопряжения микросхем различных серий и различных производителей решаются в настоящее время путем стандартизации. Стандарты в цифровой электронике в настоящее время разрабатываются на международном уровне. Ведущей в этой области организацией считается JEDEC (*Joint Electron Device Engineering Council*), являющаяся органом стандартизации при альянсе производителей электроники EIA (*Electronic Industries Alliance*).

В качестве примера действующего стандарта рассмотрим таблицу 3.3, где приведены допустимые уровни напряжений, установленные JEDEC (www.jedec.org) для ряда микросхем с напряжением питания 3,3В:

Таблица 3.3

Допустимые уровни напряжений для микросхем с напряжением питания $V_{cc} = 3,3$

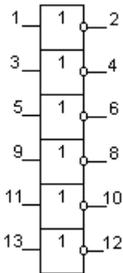
В

Обозначение	Минимальное значение (В)	Максимальное значение (В)
Семейство LVT (BiCMOS)		
VOH	2,4	
VOL		0,4
VIH	2	$V_{cc}+0,3$
VIL	-0,3	0,8
Семейство ALVC (CMOS)		
VOH	$V_{cc}-0,2$	
VOL		0,2
VIH	2	$V_{cc}+0,3$
VIL	-0,3	0,8

Из таблицы видно, что у любого семейства минимальное значение выходного напряжения высокого уровня VOH больше минимального значения входного напряжения высокого уровня VIH, а максимальное значение выходного напряжения низкого уровня VOL меньше максимального значения входного напряжения низкого уровня VIL. Эти условия необходимы для сопряжения микросхем, когда выход одной из них должен подключаться ко входам других.

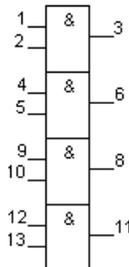
4. МАЛАЯ ЛОГИКА

Все простейшие (базовые) логические элементы: И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ, ИСКЛЮЧАЮЩЕЕ ИЛИ, входящие в наборы интегральных схем, считаются малыми интегральными схемами. Их выбор очень широк; наборы микросхем, как правило, содержат элементы с различным количеством входов и различной нагрузочной способностью. Для изготовления малых интегральных схем обычно используются корпуса с 14 выводами. В одном корпусе такой микросхемы помещается от одного до шести базовых логических элементов. На рис. 4.1- 4.4 показаны некоторые микросхемы TTL-логики, выполняющие простейшие логические преобразования.



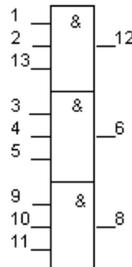
1533ЛН1

(шесть элементов НЕ)



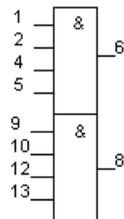
1533ЛИ1

(четыре элемента И)



1533ЛИЗ

(три элемента И)

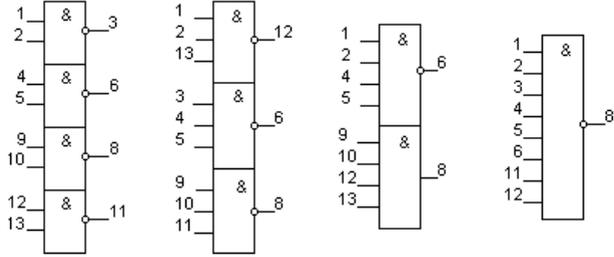


1533ЛИ6

(два элемента И)

4И)

Рис. 4.1. Логические элементы НЕ и элементы И



1533ПА3

1533ПА4

1533ПА1

1533ПА2

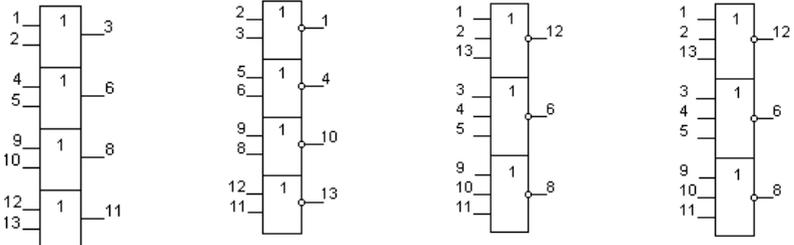
(четыре элемента И-НЕ)

(три элемента 3И-НЕ)

(два элемента 4И-НЕ)

элемент 8И-НЕ

Рис. 4.2. Элементы И-НЕ



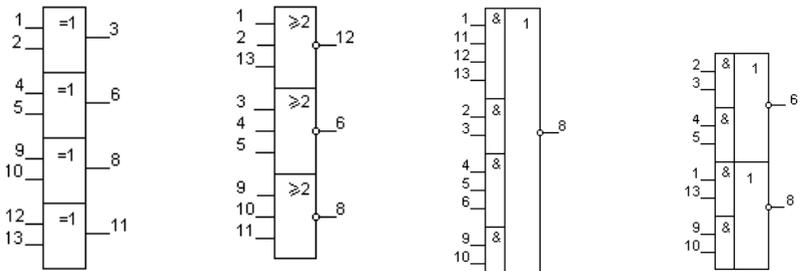
1533ЛЛ1 (четыре
элемента 2ИЛИ)

1533ЛЕ1 (четыре
элемента 2ИЛИ-НЕ)

1533ЛЕ4 (три
элемента 3ИЛИ-НЕ)

1533ЛЕ7 (два
элемента 5ИЛИ-НЕ)

Рис. 4.3. Элементы ИЛИ-НЕ



1533ЛЛ5 (четыре
элемента

533ЛЛ3 (три
трехходовых

530ЛР9 (элемент

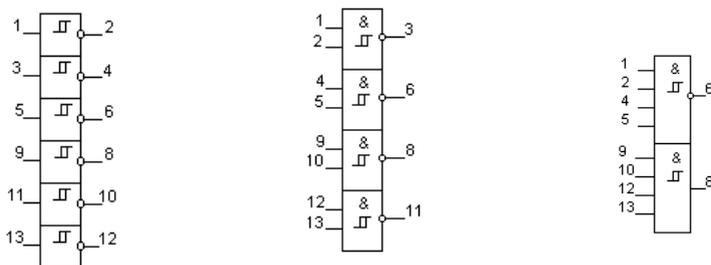
530ЛР11 (два

ИСКЛЮЧАЮЩЕЕ ИЛИ)	мажоритарных элемента с инверсией)	4-2-3-2И-4ИЛИ-НЕ)	элемента 2-2И-2ИЛИ-НЕ)
---------------------	--	-------------------	---------------------------

Рис. 4.4. Элементы с логикой ИЛИ, ИЛИ-НЕ, ИСКЛЮЧАЮЩЕЕ ИЛИ, мажоритарной логикой и логикой И-ИЛИ-НЕ

Логические элементы с повышенной помехоустойчивостью, имеющие передаточную характеристику в виде петли гистерезиса (рис. 3.2), строятся по схеме *триггера Шмитта* - усилителя с положительной обратной связью. При напряжении питания $E=5В$ напряжения переключения триггера Шмитта: $U_n^0 \approx 1,7В$, $U_n^1 \approx 0,9В$. Ширина петли гистерезиса - 0,8 В. Элементы могут выполнять функции инвертора или многоходового логического элемента И-НЕ. В стандарте МЭК и в отечественном ГОСТ элементы с гистерезисной характеристикой изображаются также как обычные элементы с добавлением символа \square (рис. 4.5.).

Триггеры Шмитта обычно используются при соединении логических схем через длинные линии, где наиболее вероятно появление различных помех.



1533ТЛ2 (шесть триггеров Шмитта с инверсией)

1533ТЛ3 (четыре триггера Шмитта с логикой 2И-НЕ)

1533ТЛ1 (два триггера Шмитта с логикой 4И-НЕ)

Рис. 4.5. Примеры TTL-триггеров Шмитта

5. БУФЕРНЫЕ ЭЛЕМЕНТЫ И ШИННЫЕ ДРАЙВЕРЫ

. Наряду с логическими элементами все серии микросхем обеспечивают широкий выбор буферных и разрешающих элементов, предназначенных для формирования и усиления электрических сигналов. Схемы отличаются от обычных логических элементов измененными выходными каскадами, допускающими подключение к ним различных элементов индикации, контроля и управления.

Схемы с открытым коллектором

Элементы *открытым коллектором* имеют упрощенный выходной каскад, в котором отсутствует цепь нагрузки. Для формирования выходного сигнала цепь нагрузки

подсоединяется к выходу элемента, как это показано пунктиром на схеме элемента с открытым коллектором TTL-логики (рис.6.1).

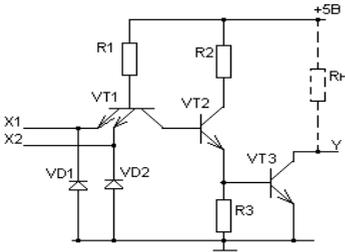


Рис. 6.1. Схема TTL-элемента И-НЕ с открытым коллектором

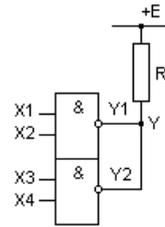


Рис. 6.2. Соединение элементов с открытым коллектором по схеме "монтажное И"

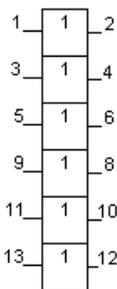
В качестве нагрузки схемы можно использовать различные индикаторы, реле и др. устройства. Элементы TTL с открытым коллектором допускают даже подключение нагрузки к отдельному источнику питания. Специальные «высоковольтные» схемы выдерживают напряжения на нагрузке до 30В.

К одной общей нагрузке допускается подсоединять выходы нескольких элементов с открытым коллектором, например, как это сделано для двух элементов И-НЕ на рис. 6.2. Для обычных элементов TTL такое соединение недопустимо из-за перегрузки выходных транзисторов..

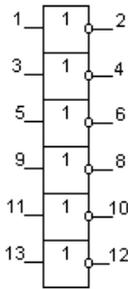
При объединении нагрузки логическая единица на выходе появляется только в случае, когда это состояние разрешено одновременно двумя элементами. Для схемы рис.6.2 такая ситуация возможна только при $Y1=Y2=1$, что соответствует логической функции схемы в виде логического произведения: $Y = Y1 \cdot Y2$. По этой причине схемное объединение выходов элементов с открытым коллектором получило название "монтажное И".

Выходы элементов с открытым коллектором отмечаются на принципиальных схемах специальным значком \diamond .

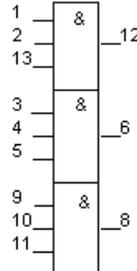
Схемы с открытым коллектором могут выполнять различные логические операции либо просто играют роль буферных усилителей (рис.6.3).



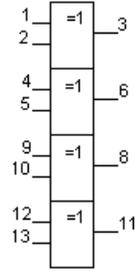
1533ПП17 (шесть буферных)



1533ЛН2 (шесть элементов НЕ с)



1533ЛИ4 (три элемента ЗИ с)



1533ЛП12 (четыре элемента)

элементов с
открытым
коллектором)

открытым
коллектором)

открытым
коллектором)

ИСКЛЮЧАЮЩЕЕ
ИЛИ с открытым
коллектором)

Рис. 6.3. Элементы с открытым коллектором

Схемы с тремя выходными состояниями

На рис. 6.4 изображена схема инвертора с тремя выходными состояниями. Инвертор имеет дополнительный вход разрешения по выходу *EO* (*Enable Output*).

В схему инвертора стандартной TTL-логики здесь добавлены диод *VD2* и инвертор *DD1*. Если на входе разрешения *EO* присутствует единичный сигнал единица, то на выходе *DD1* формируется логический ноль и потенциал коллектора транзистора *VT2* становится равным падению напряжения на открытом диоде *VD2*. Транзистор *VT2* запирается, что приводит к одновременному переходу в режим отсечки как транзистора *VT3* так и транзистора *VT4*. В результате выход элемента находится в так называемом *высокоимпеданском* состоянии, как бы отключенным от остальных цепей элемента. Если на вход *EO* поступает сигнал логического нуля, то схема работает как обычный инвертор, реализует функцию $Y = \bar{X}$. На условном обозначении логического элемента выход с тремя состояниями рекомендуется обозначать соответствующим значком Φ .

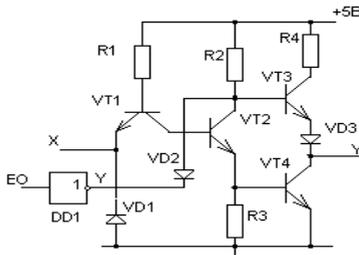
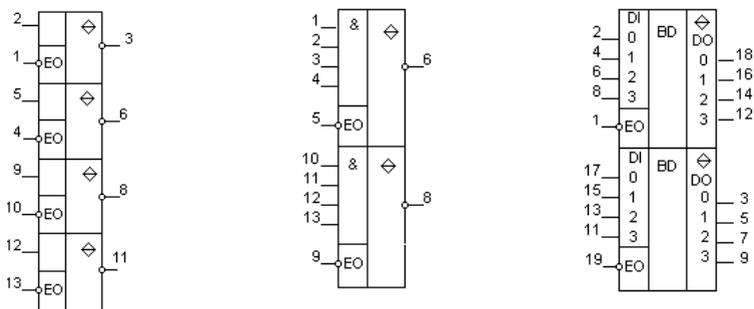


Рис. 6.4. Инвертор TTL-логики с тремя выходными состояниями

Схемы с тремя состояниями обычно играют роль буферных элементов. В ряде случаев выходы с тремя состояниями могут иметь и обычные логические элементы. Если же несколько буферных элементов в одной микросхеме имеют общие сигналы управления, то такая схема называется *шинным драйвером* (*bus driver*). Шинные драйверы выполняются четырех- или восьмиразрядными. Некоторые из них могут передавать информацию в двух направлениях. Ряд схем с тремя выходными состояниями приведен на рис. 6.5 и 6.6.

Логический вход, являющийся активным при подаче на него нулевого логического сигнала, на условном обозначении элемента обычно обозначается знаком инверсии. Таким входом у шинных формирователей рис. 6.5 и 6.6 является вход подключения выхода $\#EO$ (*Enable Output*). Знак # перед названием сигнала указывает на это его свойство.



1533ЛП8 (четыре буферных элемента с тремя выходными состояниями)

530ЛА17 (два логических элемента 4И-НЕ с тремя выходными состояниями)

1533АП5 (два четырехразрядных шинных драйвера)

Рис. 6.5. Схемы с тремя выходными состояниями

Четыре буферных элемента микросхемы К1533ЛП8 при $\#EO=0$ обеспечивают передачу сигналов со входов. В противном случае, при $\#EO=1$, выходы буферных элементов переходят в высокоимпеданное состояние. Аналогично управляются сигналами $\#EO$ логические элементы 1533ЛА17 и шинный драйвер 1533АП5. Последний объединяет в себе две четырехразрядные схемы, у которых при $\#EO=0$ данные передаются со входов *DI* (*Data Input*) на выходы *DO* (*Data Output*).

6. КОМБИНАЦИОННЫЕ ЛОГИЧЕСКИЕ СХЕМЫ

Комбинационной считается логическая схема без элементов памяти. Все выходные переменные такой схемы однозначно определяются значениями переменных входных. Связь между входными и выходными переменными задается таблицами истинности или логическими уравнениями. Далее рассматриваются наиболее распространенные комбинационные схемы, входящие как типовые элементы в состав многих серий интегральных схем TTL, ЭСЛ и CMOS-логики.

Преобразователи кодов

Преобразователи кодов осуществляют перевод двоичных чисел в другие коды, используемые на практике: двоично-десятичный код, код Грея, код "1 из N" и другие.

Двоично-десятичный код

Двоично-десятичный код позволяет отображать десятичные числа с помощью двоичных символов. Для представления каждой цифры десятичного числа используется четыре бита (тетрада), а десятичное число записывается в виде последовательности тетрад. Если каждая десятичная цифра представляется в обычном двоичном коде с весовыми коэффициентами цифр 8421, то двоично-десятичный код называется натуральным или кодом 8421. Например, запись десятичного числа 579 в натуральном двоично-десятичном коде имеет вид 12-битной последовательности: 0101 0111 1001. Поскольку для записи двоично-десятичных чисел используется только 10 комбинаций двоичных символов из 16 возможных для каждой тетрады, то двоично-десятичные числа получаются длиннее двоичных.

Преобразователи двоичного кода в двоично-десятичный обычно выполняются многоразрядными. Например, микросхема 155ПР7 (рис. 8.1) преобразует шестиразрядный двоичный код в двоично-десятичный.

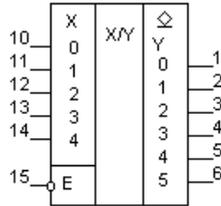


Рис. 8.1. Микросхема 155ПР7 (преобразователь двоичного кода в двоично-десятичный)

Микросхема имеет только 5 информационных входов: $X_0 \div X_4$. Подразумевается, что младший разряд X_0 двоичного кода всегда совпадает с младшим разрядом Y_0 кода двоично-десятичного и преобразование $Y_0 = X_0$ производится вне микросхемы. Выходы микросхемы $Y_0 \div Y_5$ выполнены по схеме с открытым коллектором и могут быть переведены в единичное состояние единичным сигналом на входе разрешения $\#EO$. Таблица истинности микросхемы имеет шесть входных и шесть выходных переменных.

Таблица 8.1.

Таблица истинности преобразователя в двоично-десятичный код К155ПР7

EO	X5	X4	X3	X2	X1	Y6	Y5	Y4	Y3	Y2	Y1
1	x	x	x	x	x	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	0	1	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	1	0	1	0	0	1	0	0	0
0	0	0	1	1	0	0	0	1	0	0	1
0	0	0	1	1	1	0	0	1	0	1	0
0	0	1	0	0	0	0	0	1	0	1	1
0	0	1	0	0	1	0	0	1	1	0	0
0	0	1	0	1	0	0	1	0	0	0	0
0	0	1	0	1	1	0	1	0	0	0	1
0	0	1	1	0	0	0	1	0	0	1	0
0	0	1	1	0	1	0	1	0	0	1	1
0	0	1	1	1	0	0	1	0	1	0	0
0	0	1	1	1	1	0	1	1	0	0	1
0	1	0	0	0	0	0	1	1	0	1	1
0	1	0	0	0	1	0	1	1	1	0	0
0	1	0	0	1	0	1	1	1	0	0	0
0	1	0	0	1	1	1	1	1	0	0	1
0	1	0	1	0	0	1	0	0	0	1	1
0	1	0	1	0	1	1	0	0	1	0	0

EO	X5	X4	X3	X2	X1	Y6	Y5	Y4	Y3	Y2	Y1
0	1	0	1	1	0	1	0	0	0	0	0
0	1	0	1	1	1	1	0	0	0	0	1
0	1	1	0	0	0	1	0	0	0	1	1
0	1	1	0	0	1	1	0	1	1	0	0
0	1	1	0	1	0	1	0	1	0	0	0
0	1	1	0	1	1	1	0	1	0	0	1
0	1	1	1	0	0	1	0	1	0	1	1
0	1	1	1	0	1	1	0	1	1	0	0
0	1	1	1	1	0	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	1

Обратное преобразование двоично-десятичного кода в двоичный выполняет микросхема 155ПР6, построенная по аналогичному принципу.

Код Грея

Код Грея отличается от обычного двоичного кода тем, что при изменении любого числа на единицу изменяется только один его двоичный разряд. Код Грея не позволяет выполнять арифметические операции и его используют обычно только при передаче информации. Для представления любого числа в коде Грея необходимо столько же бит, что и для обычной двоичной записи. Поэтому число входов в преобразователе двоичного кода в код Грея всегда равно числу выходов. В табл. 8.2 в коде Грея представлены все десятичные числа от нуля до 15.

Таблица 8.2.

Таблица истинности для преобразователя двоичного кода в код Грея

Десятичная запись	Двоичный код				Код Грея			
	X3	X2	X1	X0	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Из таблицы видно, что выходные Y0, Y1, Y2, Y3 и входные X0, X1, X2, X3 переменные преобразователя можно связать следующими уравнениями:

$$Y_3 = X_3, \quad Y_2 = X_3 \oplus X_2, \quad Y_1 = X_2 \oplus X_1, \quad Y_0 = X_1 \oplus X_0 \quad (8.1)$$

Преобразование двоичного кода в код Грея по уравнениям (8.1) выполняет схема на логических элементах ИСКЛЮЧАЮЩЕЕ ИЛИ, изображенная на рис. 8.2.

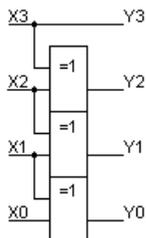


Рис. 8.2. Схема преобразования двоичного кода в код Грея

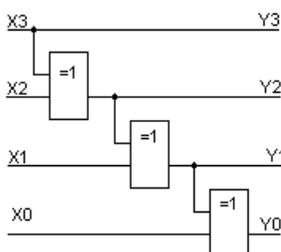


Рис. 8.3. Схема преобразователя кода Грея в двоичный код

Обратное преобразование Кода Грея в двоичный код также производится с помощью элементов ИСКЛЮЧАЮЩЕЕ ИЛИ. В этом случае переменные X_i и Y_i меняются местами и уравнения выглядят следующим образом:

$$X_3 = Y_3, \quad X_2 = Y_2 \oplus Y_3, \quad X_1 = Y_3 \oplus Y_2 \oplus Y_1, \quad X_0 = Y_3 \oplus Y_2 \oplus Y_1 \oplus Y_0 \quad (8.2)$$

Схема преобразователя кода Грея в двоичный код изображена на рис. 8.3.

Дешифраторы

Дешифратором (*Decoder*) называется преобразователь двоичного кода в код «1 из N». Код «1 из N» связывает N двоичных чисел от 0 до N-1 и N выходных переменных Y_N . Каждая выходная переменная принимает единичное значение при появлении на входе схемы двоичного числа N. Двоичные числа представляются n двоичными переменными, причем обязательно $2^n \geq N$. Например, код «1 из 8» (табл. 8.3) связывает 3 входных переменных $X_0 \div X_2$ и 8 выходных $Y_0 \div Y_7$.

Таблица 7._.

Таблица истинности дешифратора 3x8.

N	X2	X1	X0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	1	0	0
3	0	1	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	0	1	0	0	0	0
5	1	0	1	0	0	1	0	0	0	0	0
6	1	1	0	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	0

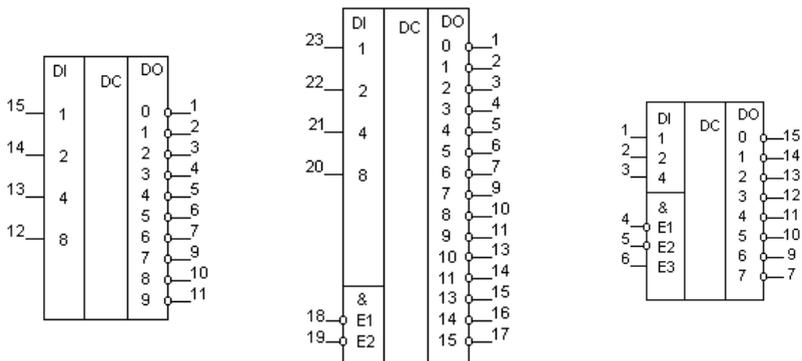
По таблице несложно представить логические функции в совершенной дизъюнктивной нормальной форме.

$$Y_0 = \overline{X_0} \cdot \overline{X_1} \cdot \overline{X_2} \cdot \overline{X_3}, \quad Y_1 = X_0 \cdot \overline{X_1} \cdot \overline{X_2} \cdot \overline{X_3}, \quad Y_2 = \overline{X_0} \cdot X_1 \cdot \overline{X_2} \cdot \overline{X_3},$$

$$Y_3 = X_0 \cdot X_1 \cdot \overline{X_2} \cdot \overline{X_3}, \quad Y_4 = \overline{X_0} \cdot \overline{X_1} \cdot X_2 \cdot \overline{X_3}, \quad Y_5 = X_0 \cdot \overline{X_1} \cdot X_2 \cdot \overline{X_3},$$

$$Y_6 = \overline{X_0} \cdot X_1 \cdot X_2 \cdot \overline{X_3}, \quad Y_7 = X_0 \cdot X_1 \cdot X_2 \cdot \overline{X_3}$$

На рис. 8.4 представлены некоторые дешифраторы в интегральном исполнении.



555ИД6 (дешифратор 4x10) 1533ИД3 (дешифратор 4x16) 1533ИД7 (дешифратор 3x8)

Рис. 7.4. Дешифраторы

Разрешающие входы #E1, #E2, E3 (Enable) позволяют отключать дешифраторы. Микросхема 1533ИД3 работает при $\overline{E1} = \overline{E2} = 0$, микросхема 1533ИД7 – при $\overline{E1} = \overline{E2} = 0, E3 = 1$. В противных случаях на всех выходах дешифраторов устанавливаются единичные сигналы.

Наиболее распространенные интегральные схемы дешифраторов приведены в табл. 7.1. Дешифраторы с количеством входных разрядов более 4 в интегральном исполнении не производятся. Для увеличения разрядности используется параллельное соединение нескольких схем. Например, на рис. 7.2 четырехходовой дешифратор с инверсными выходами построен из двух трехходовых микросхем 1533ИД7.

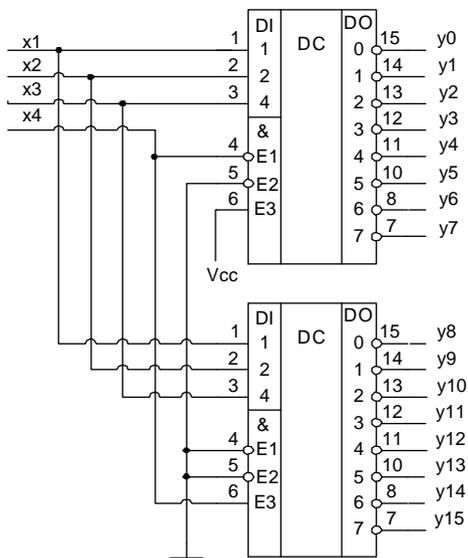


Рис. _____. Пример наращивания разрядности дешифратора

Дешифратор формирует все возможные логические произведения от входных переменных и его легко можно приспособить для формирования любых логических функций. Если логическая функция описывается уравнением в совершенной дизъюнктивной нормальной форме, то все необходимые составляющие уравнения могут быть взяты с выходов дешифратора и объединены элементами ИЛИ.

Например, функция трех переменных $y = x_1x_2x_3 \vee \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \vee x_1 \cdot \bar{x}_2 \cdot x_3$ реализуется на рис 7.5 на дешифраторе 3x8 и трехвходовом логическом элементе ИЛИ.

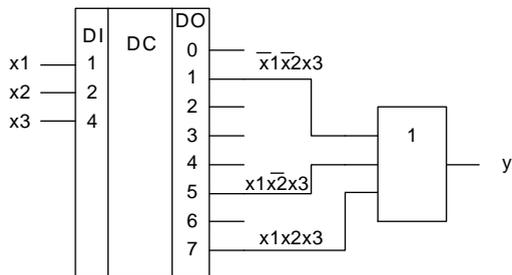


Рис. 7.5. Пример реализации функции трех переменных на дешифраторе

Дешифраторы для управления семисегментными индикаторами

Семисегментные индикаторы содержат семь светодиодов, расположенных в корпусе в определенном порядке. При этом семь сегментов индикатора (*a, b, c, d, e, f, g*), способны отобразить все десятичные цифры от 0 до 9 (рис. 7.6).

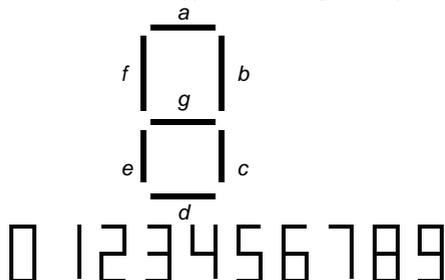


Рис. 7.6. Семисегментный индикатор

Преобразователь двоичного кода в семисегментный (семисегментный дешифратор) строится по таблице истинности, соответствующей рисунку 7.6.

Таблица 7.5.

Таблица истинности семисегментного дешифратора

N	Двоичный код				Семисегментный код						
	X3	X2	X1	X0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	0	0	0	0
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

На рис. 7.7 изображена микросхема семисегментного дешифратора 514ИД2, работающая по таблице 8.5.

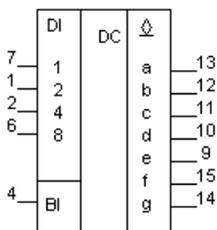


Рис. 8.7. Микросхема 514ИД2 (семисегментный дешифратор)

Схема имеет выходы с открытым коллектором, предназначенные для подключения светодиодного семисегментного индикатора с объединенными анодами. Индикаторы подключаются к выходам схемы с токоограничивающими резисторами.

Для гашения индикатора на вход *VI* необходимо подать сигнал логического нуля. Этот же вход можно использовать для регулировки яркости свечения индикатора, подавая на него импульсы напряжения различной скважности.

Шифраторы

Схема, выполняющая обратное преобразование кода “1 из N” в двоичный называется *приоритетный шифратор (coder)*. На его выходах формируется двоичное число, соответствующее старшему входу, на котором присутствует логическая единица. Значения остальных входов не имеют значения (табл. 8.4)

Таблица 8.4.

Таблица истинности приоритетного шифратора 10 x4

N	X8	X7	X6	X5	X4	X3	X2	X1	X0	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	0	0	1	x	0	0	1	0
3	0	0	0	0	0	1	x	x	x	0	0	1	1
4	0	0	0	0	1	x	x	x	x	0	1	0	0
5	0	0	0	1	x	x	x	x	x	0	1	0	1
6	0	0	1	x	x	x	x	x	x	0	1	1	0
7	0	1	x	x	x	x	x	x	x	1	0	0	0
8	1	x	x	x	x	x	x	x	x	1	0	0	1

Знаком “x” в таблице отмечено произвольное состояние переменной.

На рис. 8.5 изображена микросхема K555ИВЗ, выполняющая функции двоично-десятичного приоритетного шифратора с инверсными входами и инверсными выходами.

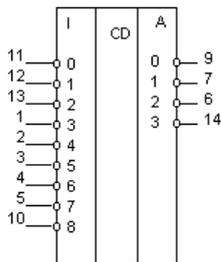


Рис. 8.5. Микросхема 555ИВЗ (двоично-десятичный приоритетный шифратор)

Мультиплексоры

Цифровая схема, имеющая много информационных входов и один выход, выполняющая функции многопозиционного переключателя (коммутатора) называется *мультиплексором*. Каждый информационный вход мультиплексора имеет свой номер (адрес). Двоичное число, поданное на адресные входы мультиплексора, задает номер входа, соединяемого с выходом элемента. Число адресных входов n и число информационных входов N мультиплексора однозначно связаны соотношением: $N = 2^n$. Например, мультиплексор с тремя адресными входами должен иметь 8 входов информационных (рис. 8.8)

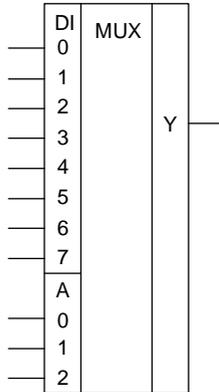


Рис. 8.8. Условное обозначение мультиплексора 8х1

В таблице истинности восьмивходового мультиплексора (табл. 8.6) значения $X_0 \div X_7$ соответствуют состояниям его восьми информационных входов.

Таблица 8.6.

Таблица истинности восьмивходового мультиплексора

Номер адресного входа	Адрес	Состояние выхода
N	A2 A1 A0	Y
0	0 0 0	X0
1	0 0 1	X1
2	0 1 0	X2
3	0 1 1	X3
4	1 0 0	X4
5	1 0 1	X5
6	1 1 0	X6
7	1 1 1	X7

По таблице однозначно записывается логическая функция мультиплексора:

$$Y = X_0 \cdot \overline{A_0} \cdot \overline{A_1} \cdot \overline{A_2} \vee X_1 \cdot \overline{A_0} \cdot \overline{A_1} \cdot A_2 \vee X_2 \cdot \overline{A_0} \cdot A_1 \cdot \overline{A_2} \vee X_3 \cdot \overline{A_0} \cdot A_1 \cdot A_2 \vee X_4 \cdot A_0 \cdot \overline{A_1} \cdot \overline{A_2} \vee X_5 \cdot A_0 \cdot \overline{A_1} \cdot A_2 \vee X_6 \cdot A_0 \cdot A_1 \cdot \overline{A_2} \vee X_7 \cdot A_0 \cdot A_1 \cdot A_2 \quad (8.3)$$

Мультиплексор считается универсальной схемой, позволяющей реализовать любую функцию алгебры логики, представленную в совершенной дизъюнктивной нормальной форме. Мультиплексор имеющий n адресных входов, в общем случае позволяет воспроизвести любую булеву функцию от $n+1$ переменных. При этом входные переменные $x_1 \dots x_n$ подаются на адресные входы мультиплексора, а на информационных входах устанавливается либо константа 0, либо константа 1, либо входная переменная x_{n+1} , либо её инверсия. На рис. 8.9 на основе мультиплексора с тремя адресными входами реализуется функция алгебры логики $y = x_3 \overline{x_2} \overline{x_1} \vee \overline{x_3} x_2 x_1 \vee x_2 \overline{x_1} \vee \overline{x_3} x_2 x_1 \overline{x_0}$.

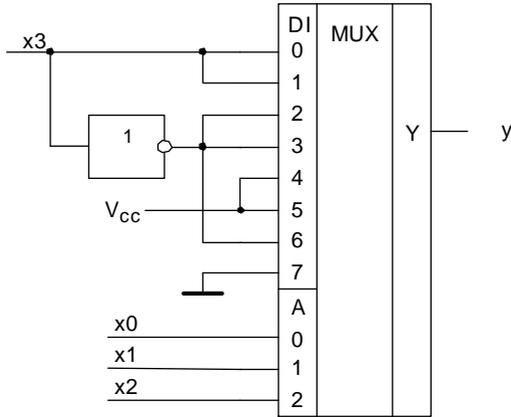
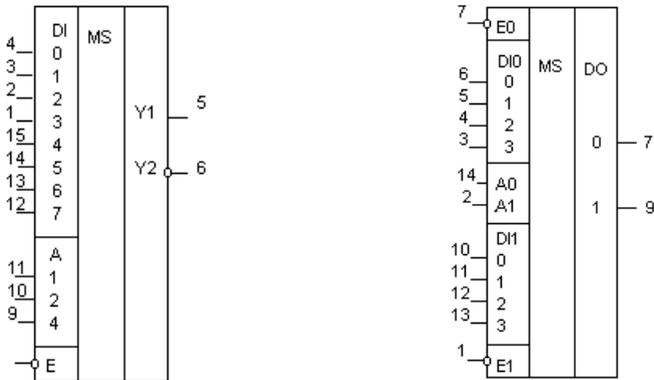


Рис. 8.9. Пример реализации логической функции на мультиплексоре
 Популярные микросхемы мультиплексоров изображены на рис. 8.9



1531КП7 (восьмивходовой мультиплексор)

1533КП2 (сдвоенный четырехвходовой мультиплексор)

Рис. 8.8. Мультиплексоры

Микросхема 1531КП7 имеет вход разрешения #E. Мультиплексор работает при #E=0. В случае #E=1 на выходах микросхемы устанавливаются сигналы Y1=0, Y2=1.

В схеме 1533КП2 объединены два четырехвходовых мультиплексора. Они имеют общие адресные входы A0, A1 и отдельные выходы DO0, DO1.

Демультимплексор решает обратную задачу - переключение одного входного сигнала на несколько выходов. Схема имеет один информационный вход, n адресных входов и $N = 2^n$ выходов. По сути, эту функцию выполняет обычный дешифратор $n \times N$, снабженный входом разрешения. Например, в микросхеме дешифратора 3×8 типа К153ЗИД7 (рис. 8.4) предусмотрено три входа разрешения: два инверсных 4, 5 и один прямой 6.

В режиме демультимплексирования один из разрешающих входов используют как информационный. На остальные входы необходимо подключить постоянные логические сигналы 0 или 1, разрешающие работу схемы.

Компараторы

Компаратор (*Comparator*) предназначен для сравнения двоичных чисел. Три выхода компаратора соответствуют трем возможным результатам сравнения: =, > или <. Таблица истинности простейшего одноразрядного компаратора имеет две входных X и Y переменные и три выходных $F1(X=Y)$, $F2(X>Y)$ и $F3(X<Y)$ (табл. 3.6)

Таблица 8.7.

Таблица истинности одноразрядного компаратора

X	Y	F1(X=Y)	F2(X>Y)	F3(X<Y)
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

В соответствии с таблицей получаем уравнения, описывающие работу устройства: $F1 = X \oplus Y$, $F2 = X \cdot \bar{Y}$, $F3 = \bar{X} \cdot Y$. Соответствующая уравнениям схема изображена на рис. 8.9.

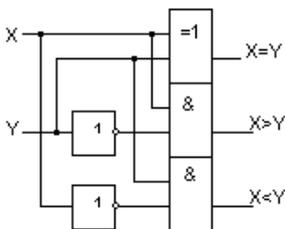


Рис. 8.9. Одноразрядный компаратор

При сравнении многоразрядных чисел сначала сравнивают значения старших разрядов двух чисел. Если они различны, то результат сравнения определяется. При равенстве старших разрядов сравнивают следующие, более младшие, и т.д. На рис. 8.10 изображен четырехразрядный компаратор с возможностью увеличения разрядности.

Для наращивания разрядности в схеме предусмотрено три входа каскадирования $Y>X$, $Y=X$, $Y<X$. При сравнении четырехразрядных чисел на вход каскадирования $Y=X$

необходимо подать сигнал высокого уровня. При большем числе разрядов используют несколько микросхем, соединенных последовательно: выходы микросхем, сравнивающих младшие разряды чисел, соединяются с соответствующими входами каскадирования микросхем, сравнивающих старшие разряды

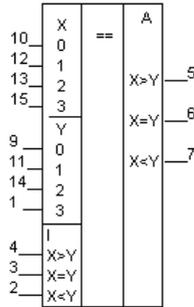


Рис. 8.10. Микросхема K1533CP1 (четырёхразрядный компаратор)

. Таблица истинности четырёхразрядного компаратора (табл. 8.8) кроме входных переменных Y и X содержит переменные $Y>X$, $Y=X$, $Y<X$.

Таблица 8.8.

Таблица истинности компаратора K1533CP1

Y,X				Входы			Выходы		
				Y>X	Y=X	Y<X	Y>X	Y=X	Y<X
Y3=X3	x	x	x	x	x	x	1	0	0
Y3<X3	x	x	x	x	x	x	0	0	1
Y3=X3	Y2>X2	x	x	x	x	x	1	0	0
Y3=X3	Y2<X2	x	x	x	x	x	0	0	1
Y3=X3	Y2=X2	Y1>X1	x	x	x	x	1	0	0
Y3=X3	Y2>X2	Y1<X1	x	x	x	x	0	0	1
Y3=X3	Y2<X2	Y1=X1	Y0>X0	x	x	x	1	0	0
Y3=X3	Y2=X2	Y1>X1	Y0<X0	x	x	x	0	0	1
Y3=X3	Y2>X2	Y1<X1	Y0=X0	1	0	0	1	0	0
Y3=X3	Y2<X2	Y1=X1	Y0>X0	0	0	1	0	0	1
Y3=X3	Y2=X2	Y1>X1	Y0<X0	x	1	x	0	1	0
Y3=X3	Y2>X2	Y1<X1	Y0=X0	1	0	1	0	0	0
Y3=X3	Y2<X2	Y1=X1	Y0=X0	0	0	0	1	0	1

Схемы контроля четности

Схема контроля четности анализирует количество единиц в двоичном числе. Сигнал четности формируется на выходе при четном количестве единиц. Таблица истинности схемы контроля четности обычно содержит две выходные переменные: Y1 и Y2, соответствующие четному и нечетному количеству единиц в числе. Количество входных переменных соответствует разрядности схемы (табл. 8.9).

Таблица 8.9.

Таблица истинности трехразрядной схемы контроля четности

X2	X1	X0	Y1 (чет)	Y2 (нечет)
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

В соответствии с таблицей логическая функция схемы контроля четности легко выражается с помощью операции ИСКЛЮЧАЮЩЕЕ ИЛИ: $Y1 = X1 \oplus X2 \oplus X3$, $Y2 = \overline{Y1}$.

Типовая восьмиразрядная схема контроля четности изображена на рис. 8.11.

При $OE=0$ схема осуществляет контроль четности. При четном числе единиц на входах DI на выходах: $PE=1$, $PO=0$.

При $OE=1$ схема контролирует нечетность суммы. В этом случае при четном количестве единиц на входах DI имеем $PE=0$, $PO=1$.

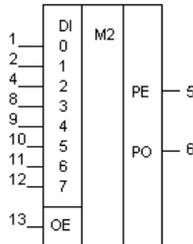


Рис. 8.11. Микросхема К1533ИП5 (восьмиразрядная схема контроля четности)

Сумматоры

Полусумматор

Сумматор предназначен для сложения двоичных чисел. Простейшим сумматором является схема осуществляющая сложение двух одноразрядных чисел, известная под названием *полусумматор*. Поскольку результат сложения двух одноразрядных чисел может быть двухразрядным: $0+0=0$, $0+1=1$, $1+1=10$, то полусумматор имеет два входа и два выхода: сумма S и перенос C (*Carry*).

Таблица 8.10.

Таблица истинности полусумматора

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

По таблице находим: $C = A \cdot B$, $S = A \oplus B$ Схема полусумматора, соответствующая приведенным логическим функциям, приведена на рис. 8.12.

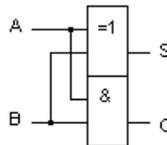


Рис. 8.12. Полусумматор

Одноразрядный полный сумматор

При сложении многоразрядных чисел в каждом разряде необходимо учитывать сигнал переноса из младшего разряда. Фактически необходимо всегда складывать три одноразрядные переменные: два слагаемых и сигнал переноса. Эту операцию выполняет *одноразрядный полный сумматор*. В таблице истинности одноразрядного полного сумматора (табл. 8.11) три входные переменные: A_i , B_i и перенос C_i и две выходные: сумма S_i и перенос в старший разряд C_{i+1} .

Таблица 8.11.

Таблица истинности одноразрядного полного сумматора

A_i	B_i	C_i	C_{i+1}	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Уравнения, соответствующие таблице, имеют вид:

$$S_i = A_i \oplus B_i \oplus C_i, \quad C_{i+1} = A_i \cdot B_i \vee (A_i \oplus B_i) \cdot C_i \quad (8.4)$$

Простая схема, построенная по уравнениям (8.4), изображена на рис. 8.13. Микросхема 555ИМ5, содержащая два одноразрядных полных сумматора, представлена на рис. 8.14.

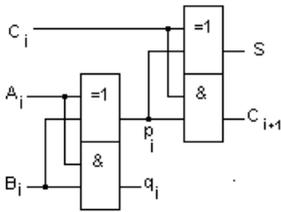


Рис. 8.13. Схема одноразрядного полного сумматора

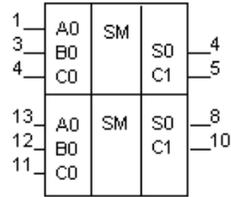


Рис. 8.14. Микросхема 555ИМ5 (два одноразрядных сумматора)

Многоразрядный сумматор

Многоразрядные сумматоры, объединяют в себе несколько одноразрядных схем. При этом выходы переноса младших разрядов соединяются со входами переноса старших. На рис. 8.15 приведен пример построения двухразрядного сумматора из двух одноразрядных. Функции полного двухразрядного сумматора выполняет микросхема 155ИМ2 (рис. 8.16).

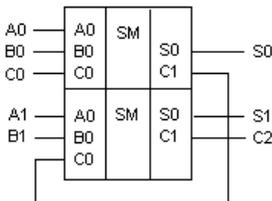


Рис. 8.15. Схема двухразрядного полного сумматора

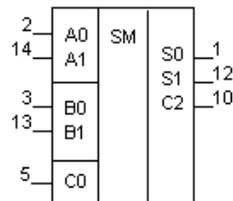


Рис. 8.16. Микросхема 155ИМ2 (двухразрядный сумматор)

Время выполнения операции в многоразрядном сумматоре по схеме рис. 8.15 существенно превышает время сложения в сумматоре одноразрядном из-за последовательного распространения сигнала переноса в рассмотренной схеме (*Ripple Carry*). Для уменьшения времени вычислений в многоразрядных схемах используют схемы параллельного переноса (*Carry look-ahead*). При этом все сигналы переноса вычисляются непосредственно по значениям входных переменных. Согласно (4.4) в каждом разряде многоразрядного числа сигнал переноса имеет две составляющие:

$$C_{i+1} = A_i \cdot B_i \vee (A_i \oplus B_i) \cdot C_i = g_i \vee q_i \quad (8.5)$$

Первая составляющая g_i , называемая функцией генерацией переноса, формируется из входных сигналов A_i и B_i . Вторая q_i , называемая функцией распространения переноса, связывает перенос из младшего разряда с переносом в следующий - более старший.

Из формулы 8.5 следует:

$$\begin{aligned}
 C_1 &= g_0 + q_0 C_0, \\
 C_2 &= g_1 + q_1 C_1 = g_1 + q_1 g_0 + q_1 q_0 C_0, \\
 C_3 &= g_2 + q_2 C_2 = g_2 + q_2 g_1 + q_2 q_1 g_0 + q_2 q_1 q_0 C_0
 \end{aligned}
 \tag{8.6}$$

и т.д.

Хотя полученные выражения достаточно сложны, время формирования сигнала переноса сокращается, т.к. все вычисления производятся одновременно. Схема, реализующая вычисления по формуле 8.6 известна под названием *схема ускоренного переноса*. Она может входить в состав многоразрядного сумматора или выпускается в виде отдельной микросхемы. На рис. 8.17 изображена микросхема 155ИИМ3, выполняющая функции четырехразрядного полного сумматора с параллельным переносом. Четырехразрядная схема ускоренного переноса 153ЗИП4 изображена на рис. 8.18.

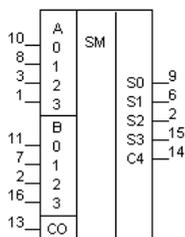


Рис. 8.17. Микросхема 155ИИМ3
(четырёхразрядный полный сумматор с
параллельным переносом)

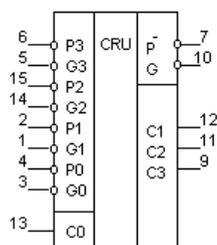


Рис. 8.18. Микросхема 153ЗИП4 (схема
ускоренного переноса)

Любой многоразрядный сумматор можно использовать для сложения чисел со знаком. При этом знаковым считается старший разряд числа, а отрицательные числа представляются в дополнительном коде.

Арифметико-логические устройства

Арифметико-логическим устройством называется универсальная схема, способная выполнять различные арифметические и логические операции с двумя переменными. Обычно арифметико-логическое устройство осуществляет преобразования четырехразрядных переменных и имеет возможность наращивания разрядности. Типовая микросхема четырехразрядного арифметико-логического устройства изображена на рис. 8.19.

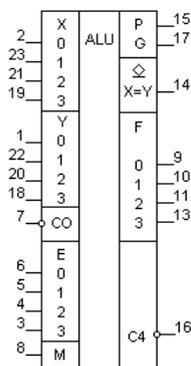


Рис. 8.19. Микросхема 1533ИП3 (четырёхразрядное АЛУ)

Кроме входов двух четырехразрядных операндов X и Y микросхема имеет вход переноса $\#CO$, четырехразрядный вход задания кода операции E , вход выбора режима (арифметический/логический) M , выход переноса $C4$, выходы для организации параллельного переноса P и G , выход равенства переменных $X=Y$.

В зависимости от состояния входа выбора режима M схема выполняет либо 16 логических, либо 16 арифметических операций. Все логические операции выполняются поразрядно. Выход равенства переменных имеет схему с открытым коллектором. Выходы распространения и генерации переноса позволяют при наращивании разрядности использовать схему быстрого переноса 1533ИП4 (рис. 8.18).

Работа микросхемы описывается таблицей режимов (табл. 8.12). В приведенной таблице столбец $M=1$ соответствует логическому, а $M=0$ – арифметическому режиму работы схемы.

Таблица 8.12.

Таблица режимов четырехразрядного арифметико-логического устройства

E3	E2	E1	E0	M=1	M=0
0	0	0	0	\bar{X}	$X + CO$
0	0	0	1	$\overline{X \vee Y}$	$X \vee Y + CO$
0	0	1	0	$\bar{X} \cdot Y$	$X \vee Y + CO$
0	0	1	1	0	$-1 + CO$
0	1	0	0	$\overline{X \cdot Y}$	$X + (X \cdot \bar{Y}) + CO$
0	1	0	1	\bar{Y}	$(X \vee Y) + (X \cdot \bar{Y}) + CO$
0	1	1	0	$X \oplus Y$	$X - Y - 1 + CO$
0	1	1	1	$X \cdot \bar{Y}$	$X \cdot \bar{Y} - 1 + CO$

E3	E2	E1	E0	M=1	M=0
1	0	0	0	$\overline{X \vee Y}$	$X + (X \cdot Y) + CO$
1	0	0	1	$\overline{X \oplus Y}$	$X + Y + CO$
1	0	1	0	Y	$f = 1 / \Delta t$
1	0	1	1	$X \cdot Y$	t_{PLH}
1	1	0	0	1	$X + X + CO$
1	1	0	1	$X \vee \overline{Y}$	$(X \vee Y) + X + CO$
1	1	1	0	$X \vee Y$	$(X \vee \overline{Y}) + X + CO$
1	1	1	1	X	$X + CO$

7. ТРИГГЕРЫ

Триггер является простейшей ячейкой памяти, способной при включенном питании хранить один бит информации. Различные схемы триггеров отличаются способами записи информации, логикой работы и элементной базой. Наиболее распространенными в настоящее время являются: RS-триггеры, JK-триггеры, D-триггеры и T-триггеры.

RS-триггеры

Триггер типа RS имеет два информационных входа: вход установки S (Set) и вход сброса R (Reset). Схема триггера строится на двух элементах ИЛИ-НЕ или И-НЕ, охваченных перекрестными обратными связями. На рис. 7.1 изображена схема триггера на элементах ИЛИ-НЕ.

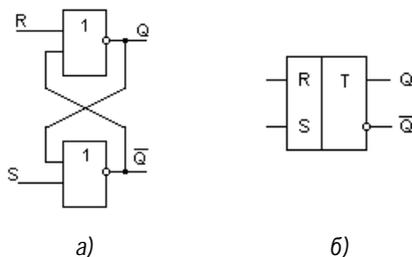


Рис. 7.1. RS-триггер на элементах ИЛИ-НЕ: схема (а) и условное обозначение (б)

Режимы работы триггера задаются состояниями его входов R и S . При этом выходы триггера Q и \overline{Q} изменяются в соответствии с табл. 7.1:

Таблица 7.1

Таблица переходов триггера типа RS на элементах ИЛИ-НЕ

Режим	R	S	Q	\overline{Q}
-------	---	---	---	----------------

запись 1	0	1	1	0
запись 0	1	0	0	1
хранение	0	0	не изменяется	
запрещено	1	1	0	0

Последнее состояние считается запрещенным, так как оно не сохраняется при переводе триггера в режим хранения; триггер попадает в так называемое неопределенное состояние, когда на его выходах сигналы с равной вероятностью могут быть как нулевыми, так и единичными. По этой причине запрещенный режим использовать не рекомендуется.

На рис. 7.2 изображен RS-триггер на элементах И-НЕ. В качестве входных сигналов этого триггера используются инверсные переменные #R и #S.

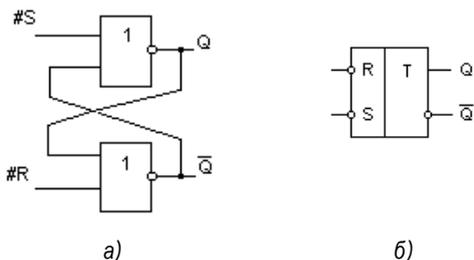


Рис. 7.2. RS-триггер на элементах И-НЕ а) схема б) условное обозначение

Работа триггера описывается несколько измененной таблицей переходов (табл.7.2).

Таблица 7.2.

Таблица переходов RS-триггера на элементах И-НЕ

Режим	#R	#S	Q	\bar{Q}
запись 1	1	0	1	0
запись 0	0	1	0	1
хранение	1	1	не изменяется	
запрещено	0	0	1	1

В любом случае выходная переменная Q зависит не только от состояния входов R и S, но и от предыдущего состояния триггера Q^* . Если считать состояние триггера в запрещенном режиме безразличным, то таблицы истинности 7.1 и 7.2. равноценны. В карте Карно (рис. 7.3), соответствующей таблицам, запрещенное состояние обозначено символом "x".

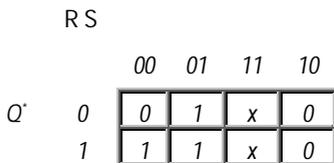


Рис. 7.3. Карта Карно для RS-триггера

Зависимость выходной переменной Q от входных R и S называется *характеристическим уравнением* триггера. На основании карты рис. 7.3 характеристическое уравнение RS-триггера принимает вид:

$$Q = S \vee Q^* \cdot \bar{R} \text{ при } S \cdot R = 0. \quad (7.1)$$

RS-триггер может быть выполнен *синхронным*. В этом случае он будет реагировать на входные сигналы только в определенные моменты времени. Эти моменты задаются с помощью дополнительного сигнала синхронизации C (*Clock*). Схема такого триггера показана на рис. 7.4

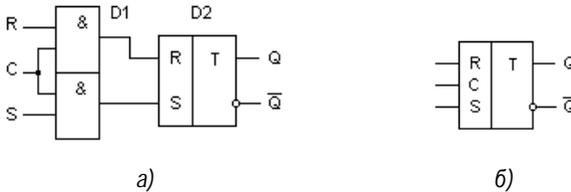


Рис. 7.4. Синхронный RS-триггер а) схема б) условное обозначение

При $C=0$ на входах триггера $D2$ переменные $R=S=0$ и триггер $D2$ находится в режиме хранения. При $C=1$ схема работает как обычный RS-триггер.

Синхронный двухступенчатый RS-триггер *типа M-S (Master-Slave)* содержит два обычных последовательно включенных синхронных триггера, один из которых считается ведущим (*Master*), а другой – ведомым (*Slave*). Сигналы синхронизации для двух триггеров инверсны (рис. 7.5).

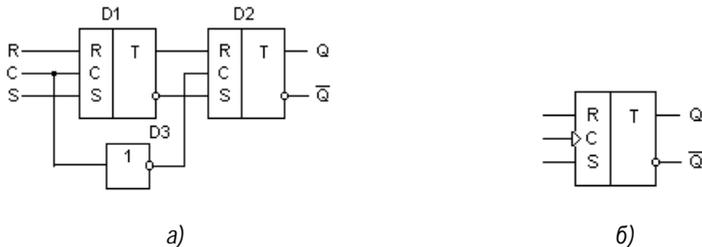


Рис. 7.5. RS-триггер типа M-S

а) схема б) условное обозначение триггера, переключающегося по заднему фронту импульса синхронизации

При единичном сигнале синхронизации C ведомый триггер $D2$ находится в режиме хранения, а ведущий $D1$ работает как обычный RS-триггер. При установке на входе синхронизации нулевого сигнала ведущий триггер $D1$ переходит в режим хранения, а ведомый $D2$ - в режим записи информации. Таким образом, изменение сигналов на выходах ведомого триггера возможно только в момент перехода импульса синхронизации из единичного состояния в нулевое (по заднему фронту импульса). Поскольку ведущий триггер работает в режиме записи всё время, пока $C=1$, то на выходах ведомого триггера устанавливаются сигналы, соответствующие состоянию RS-входов на момент заднего фронта импульса синхронизации.

RS-триггер типа M-S является основой для построения триггеров других типов.

D-триггеры

Синхронный *D-триггер* имеет вход данных *D* (*Data*) и вход синхронизации *C* (*Clock*) (рис. 7.6).

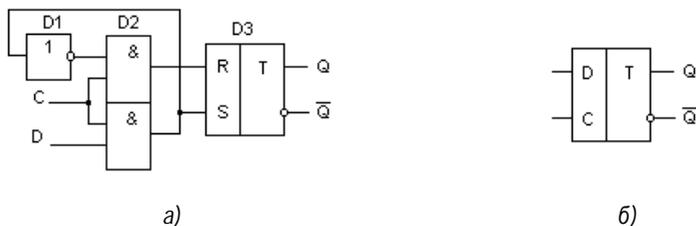


Рис.7.6. Синхронный *D-триггер* а) схема б) условное обозначение

В данной схеме при $C=0$ триггер *D3* находится в режиме хранения, а при $C=1$ записывает входную переменную *D*. Характеристическое уравнение *D-триггера* имеет вид: (7.2).

Двухступенчатый D-триггер строится на основе RS-триггера типа M-S. На рис. 7.7 с этой целью перед R-входом триггера подключен инвертор.

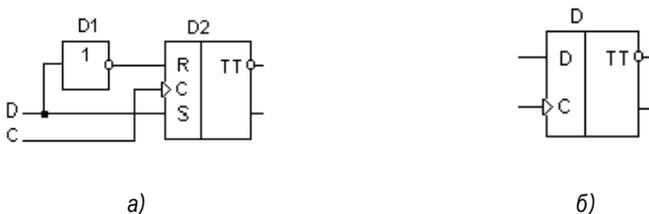


Рис.7.7. *D-триггер* типа M-S: а) схема, б) условное обозначение

Динамический D-триггер осуществляет запись информации по переднему либо по заднему фронту импульса синхронизации. В принципе эту задачу решает любая схема типа M-S, но для *D-триггеров* известны и специальные схемы. Одна из таких схем динамического *D-триггера*, управляемого передним фронтом импульса синхронизации *C*, изображена на рис. 7.8.

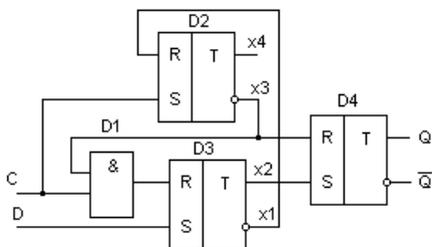


Рис. 7.8. Динамический D-триггер

При $C = 1$ независимо от состояния входа D сигналы $x_2 = x_3 = 0$ и выходной триггер D4 находится в режиме хранения. При этом: $x_1 = \overline{D}$, $x_4 = \overline{x_1} = D$ и, следовательно, один из двух входных триггеров D2, D3 находится в запрещенном режиме, которое не может быть сохранено при переходе импульса синхронизации в нулевое состояние. По заднему фронту сигнала C этот триггер из запрещенного режима переходит в режим записи и при $D=0$ получаем $x_3=1$, $x_2=0$, а при $D=1$ – $x_2=1$, $x_3=0$. В результате триггер устанавливается в состояние: $Q = D$.

Все последующие изменения сигнала D при $C = 0$ не влияют на состояния входных триггеров D2, D3. Новая информация в них будет записана только при $C = 1$, когда один из триггеров вновь перейдет в запрещенный режим.

JK-триггеры

JK-триггер всегда имеет структуру типа M-S. У триггера два информационных входа J (Jump) и K (Keer), а также необходимый при такой структуре вход синхронизации C (Clock) (рис. 7.9).

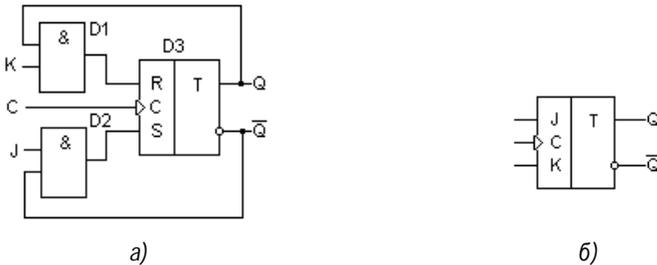


Рис. 7.9. JK-триггер а) схема б) условное обозначение

JK-триггер функционирует подобно двухступенчатому RS-триггеру, изменяя своё состояние по заднему фронту импульса синхронизации в соответствии с таблицей переходов 7.3.

Таблица 7.3

Таблица переходов JK-триггера

Режим	J	K	Q	\overline{Q}
запись 1	1	0	1	0
запись 0	0	1	0	1
хранение	0	0	не изменяется	
счетный	1	1	изменяется на противоположное	

Таблица переключений верна, если состояние JK-входов не изменяется при $C=1$. Ведущий триггер схемы может быть опрокинут только один раз и из-за наличия в схеме обратных связей не может быть возвращен в исходное состояние.

В отличие от RS-триггера режим $J=K=1$ (счетный режим) приводит к переходу JK-триггера в новое состояние, противоположное исходному.

Карта Карно JK-триггера изображена на рис. 7.10.

		JK			
		00 01 11 10			
Q*	0	0	0	1	1
	1	1	0	0	1

Рис. 7.10. Карта Карно JK-триггера

На основании рис. 7.10 получаем характеристическое уравнение JK-триггера:

$$Q = J \cdot \overline{Q}^* \vee \overline{K} \cdot Q \quad (7.3)$$

T-триггеры

T-триггер, или счетный триггер, изменяет состояние выхода на противоположное под воздействием синхронизации на входе *C*. Кроме входа синхронизации T-триггер может иметь ещё разрешающий вход *T*. Изменение состояния триггера под действием импульса синхронизации разрешается при $T=1$. Характеристическое уравнение триггера в этом случае имеет вид:

$$Q = T \cdot \overline{Q}^* \vee \overline{T} \cdot Q^* \quad (7.4)$$

Для построения счетного триггера обычно используется JK-триггер, включенный по схеме рис. 7.11.



Рис. 7.11. Счетный триггер а) схема б) условное обозначение

Объединенные входы J и K выполняют функции T-входа. При $J=K=1$ JK-триггер работает в счетном режиме (табл. 7.3).

На рис. 7.12, 7.13 изображены наиболее распространенные микросхемы TTL-триггеров.

Микросхема 155TM5 содержит две пары синхронных D-триггеров, каждая из которых имеет общий вход синхронизации *C*. Информация записывается в триггеры при $C=1$.

Каждый триггер микросхемы 1533TM2 может работать как в асинхронном так и в синхронном режимах. В асинхронном режиме триггер управляется по входам *R* и *S*. При $\overline{S} = \overline{R} = 1$ запись информации с входа *D* происходит по переднему фронту сигнала *C*.

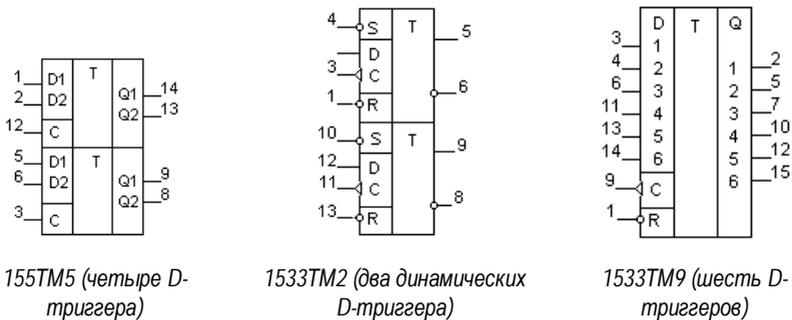


Рис. 7.12. Микросхемы D-триггеров.

Запись информации во все триггеры микросхемы 1533TM9 происходит одновременно под действием положительного фронта импульса синхронизации C . Асинхронное обнуление всех триггеров наступает при $\#R=0$.

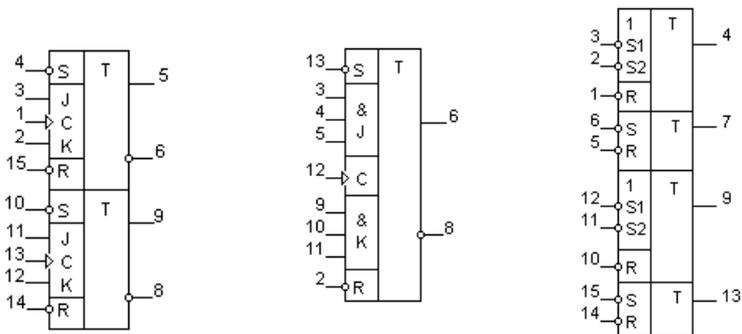


Рис. 7.13. Интегральные JK- и RS-триггеры

Запись информации со входов J и K микросхемы 1533TB9 производится по заднему фронту тактового импульса на входе C , а смена информации на входах J и K разрешается производить только при $C = 1$. У микросхемы 155TB1 входы J и входы K объединены логикой ЗИ. Асинхронные входы $\#R$ и \bar{S} $\#S$ -триггеров 1533TB9 и 155TB1 предназначены для установки триггеров соответственно в нулевое и единичное состояния. Установка производится сигналами низкого уровня.

Микросхема 1533TP2. содержит четыре RS-триггера. Каждый из триггеров имеет установочные входы $\#R$ и $\#S$. Причем два триггера имеют по два входа $\#S1$ и $\#S2$, объединенных логикой ИЛИ.

8. ПОСЛЕДОВАТЕЛЬНОСТНЫЕ ЛОГИЧЕСКИЕ СХЕМЫ

Последовательностными считаются схемы, содержащие элементы памяти. Выходные переменные такой схемы в любой момент времени зависят не только от

значений входных сигналов, но и от состояния внутренних ячеек памяти. В качестве элементов памяти в схемах последовательностного типа обычно используются различные триггеры. Набор типовых последовательностных схем очень ограничен. Обычно к ним относят различного типа счетчики и регистры, выпускаемые в интегральном исполнении в различных сериях интегральных микросхем.

Регистры

Регистром (Register) называется логическая схема, содержащая несколько триггеров, и предназначенная для временного хранения нескольких бит информации. Каждый триггер регистра способен хранить один бит (двоичный разряд) данных. В зависимости от способа записи информации различают *параллельные, последовательные и универсальные* регистры.

В *параллельном регистре* запись информации осуществляется одновременно во все триггеры. Схема параллельного регистра обычно представляет собой набор D-триггеров с объединенными входами синхронизации (рис. 9.1.).

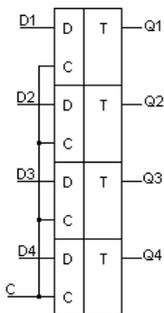


Рис. 9.1. Четырехразрядный параллельный регистр

При поступлении импульса на вход синхронизации С происходит запись информации со входов D1...D4 одновременно во все разряды регистра. В схеме могут быть использованы одноступенчатые и двухступенчатые (динамические) триггеры. В зависимости от типа используемых триггеров запись производится либо всё время существования импульса, либо по одному из его фронтов (переднему или заднему).

В *последовательном регистре* с каждым импульсом синхронизации происходит запись одного бита информации в первый триггер схемы. При этом данные, записанные в регистр ранее, переписываются из каждого триггера в последующий (сдвигаются на один разряд). По этой причине последовательный регистр иногда называют *регистром сдвига*. Схема последовательного регистра состоит из нескольких соединенных последовательно D-триггеров.

В схеме обязательно должны использоваться триггеры с динамическим входом, осуществляющие запись информации по переднему или заднему фронту импульса синхронизации. В противном случае входной сигнал за один импульс синхронизации окажется переписанным во все разряды регистра.

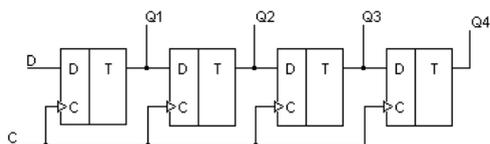


Рис.9.2. Последовательный регистр

Четырехразрядный регистр, изображенный на рисунке, может хранить четыре бита. Запись каждого бита со входа D в первый триггер происходит при поступлении импульса синхронизации на вход C. Полностью процесс заполнения регистра данными показан в таблице 9.1.

Таблица 9.1.

Таблица переходов четырехразрядного регистра сдвига

Номер импульса	Q1	Q2	Q3	Q4
1	D1			
2	D2	D1		
3	D3	D2	D1	
4	D4	D3	D2	D1

После четвертого импульса входная последовательность $D1...D4$ оказывается записанной в регистре. Таким образом, последовательный регистр осуществляет преобразование последовательной информации в параллельную.

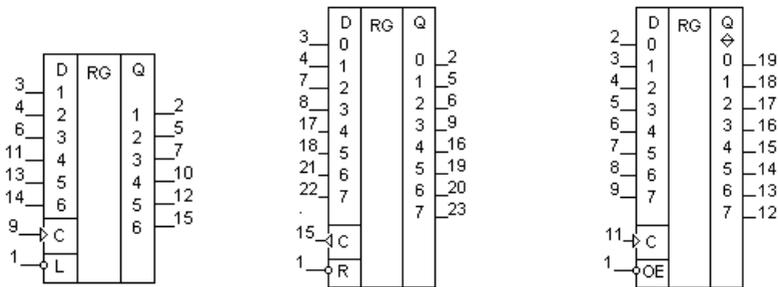
Универсальные регистры соединяют в себе свойства последовательных и параллельных схем. Они могут работать в различных режимах, обеспечивающих параллельную запись и сдвиг данных. Кроме того, они могут иметь входы обнуления (сброса) и входы управления для изменения направления сдвига: влево или вправо.

На рис. 9.3, 9.4 изображены некоторые распространенные микросхемы, выполняющие функции регистров.

Ввод информации в регистр 531ИР18 производится при $\bar{L} = 0$ по положительному фронту импульса синхронизации на входе C. При $\bar{L} = 1$ ввод запрещается.

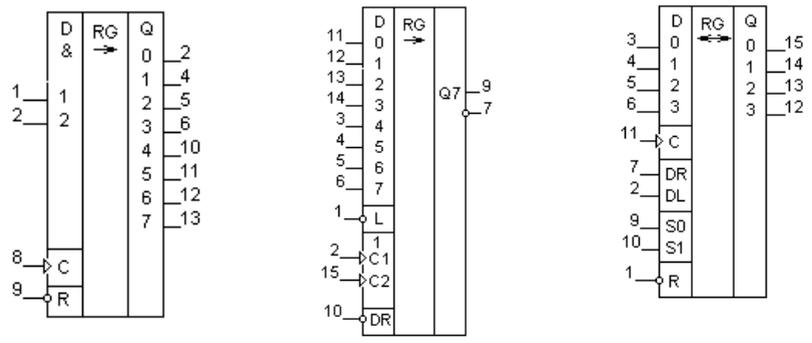
Ввод данных в регистр 555ИР35 осуществляется по заднему фронту импульса синхронизации C при $\bar{R} = 1$. При $\bar{R} = 0$ происходит асинхронное обнуление регистра.

Запись информации в регистр 1533ИР37 происходит по положительному фронту импульса C независимо от состояния входа OE , однако при $\bar{OE} = 1$ все выходы схемы переходят в высокоимпеданное состояние.



531IP18 (шестиразрядный параллельный регистр) 555IP35 (восьмиразрядный параллельный регистр) 1533IP37 (параллельный регистр с тремя состояниями выходов)

Рис. 9.3. Параллельные регистры



555IP8 (восьмиразрядный регистр сдвига) 1533IP9 (восьмиразрядный универсальный регистр) 555IP11 (четырёхразрядный универсальный регистр)

Рис. 9.4. Регистры сдвига

В регистре 555IP8 запись и сдвиг информации производятся по переднему фронту импульса синхронизации C при $\#R=1$. Записываемые данные подаются на входы $D1, D2$ объединенные логикой И. При $\#R=0$ происходит обнуление регистра.

В регистр 1533IP9 информация в параллельном виде с входов D записывается при $\#L=0$. При $\#L=1$ сдвиг информации осуществляется по переднему фронту импульсов синхронизации, поступающих на любой из входов $C1$ или $C2$. На вход $\#DR$ при этом необходимо подавать данные, предназначенные для записи в младший разряд регистра.

Регистр 555IP11 может работать в четырех различных режимах, задаваемых по входам $S1, S2$ (табл. 9.2).

Таблица режимов регистра 555ИР11

S0	S1	Режим
0	0	Хранение
0	1	Сдвиг влево
1	0	Сдвиг вправо
1	1	Параллельный ввод

Параллельный ввод данных со входов $D0..D3$ происходит по переднему фронту импульса синхронизации C при $\#R = 1$. Сдвиг информации, поступающей со входа DL (сдвиг влево) или DR (сдвиг вправо), также осуществляется по переднему фронту сигнала C . При $\#R = 0$ происходит обнуление регистра.

Счетчики

Счетчиком (*Counter*) называется логическая схема, в которой состояние выходных переменных однозначно соответствует количеству поступивших на вход импульсов. Каждая выходная переменная счетчика считается двоичным разрядом, а всё множество выходных переменных отображает результат счета в двоичной системе счисления. Чаще всего для представления результата используется обычный двоичный код. Если в процессе счета импульсов код на выходе счетчика возрастает, то счетчик называется суммирующим (Up-counter), если убывает - вычитающим (Down-counter). Счетчик, у которого направление счета может меняться в процессе работы, называется реверсивным (Up/Down counter). Основными параметрами счетчика являются разрядность и коэффициент (модуль) счета. Разрядность n определяется количеством выходов счетчика, а коэффициент счета K - общим количеством кодовых комбинаций, которые могут появиться на его выходах. В простейшем двоичном счетчике $K = 2^n$, но вместе с тем известны и широко используются схемы с произвольным значением $K < 2^n$. Например, двоично-десятичные счетчики имеют коэффициент счета 10.

Простейшим считается асинхронный суммирующий двоичный счетчик, представляющий собой цепочку последовательно соединенных счетных триггеров (рис. 9.5)

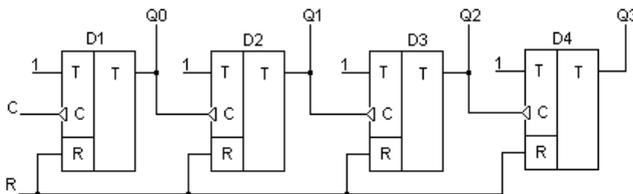


Рис. 9.5. Асинхронный суммирующий двоичный счетчик

Счетчик на рис. 9.5 четырехразрядный. На T -входы всех триггеров счетчика подан единичный разрешающий сигнал. Кроме того, каждый триггер снабжен входом сброса R . Все R -входы объединены и на них подается сигнал сброса счетчика в нулевое начальное состояние. По заднему фронту импульсов синхронизации C триггеры изменяют своё состояние в соответствии с диаграммой рис. 9.6.

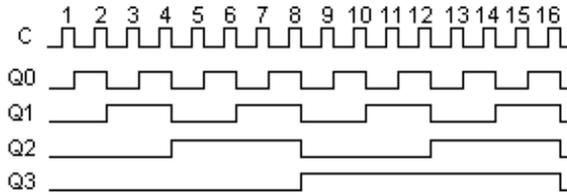


Рис. 9.6. Временные диаграммы работы двоичного счетчика

Таблица переходов счетчика (табл. 9.3) связывает количество импульсов, поступивших на его вход, и состояния выходных переменных. Таблица строится на основании рис. 9.6. при нулевых начальных состояниях выходов.

Таблица 9.3.

Таблица переходов двоичного счетчика

Импульсов	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Q0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Q1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
Q2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0
Q3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0

С приходом каждого импульса двоичный код на выходе счетчика увеличивается на единицу. Четырехразрядный двоичный счетчик считает от 0 до 15. Шестнадцатый импульс возвращает его в исходное состояние. В общем случае, n -разрядный двоичный счетчик имеет 2^n возможных кодовых комбинаций на выходах и, следовательно, его коэффициент счета равен 2^n .

В *вычитающем счетчике*, в отличие от суммирующего, при последовательном соединении триггеров используются их инверсные выходы (рис. 9.7).

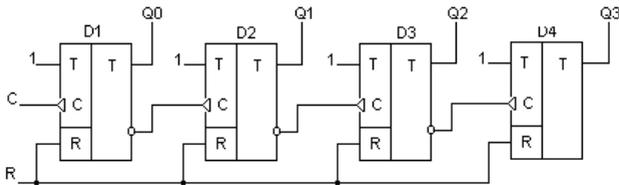


Рис. 9.7. Асинхронный вычитающий двоичный счетчик

При поступлении импульсов синхронизации C четырехразрядный код на выходах счетчика будет уменьшаться.

В *реверсивном счетчике* направление счета изменяется за счет переключения выходов триггеров. В схемах используются триггеры с прямыми и инверсными выходами, а между триггерами включается логическая схема, обеспечивающая подключение нужных

выходов. По этому принципу построена, изображенная на рис. 9.8, схема трехразрядного реверсивного счетчика.

Счетчик имеет вход суммирования CU и вход вычитания CD . Суммируемые импульсы, поступающие на вход CU , попадают на счетные входы только тех триггеров, для которых все предыдущие разряды имеют единичные значения. Вычитаемые, поступающие на вход CD , попадают на счетные входы триггеров, все предыдущие разряды которых единичные. В результате код на выходе счетчика определяется разностью числа импульсов, поступивших на его суммирующий и вычитающий входы.

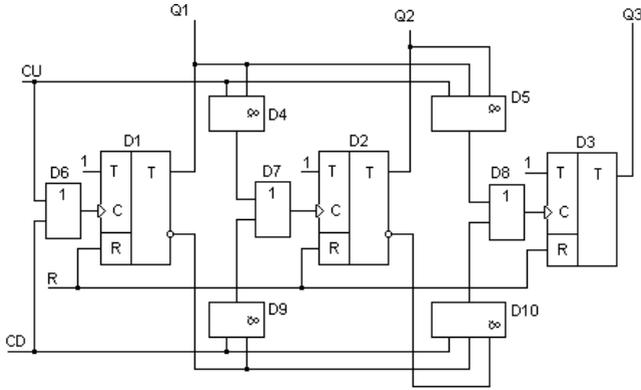


Рис. 9.8. Реверсивный счетчик

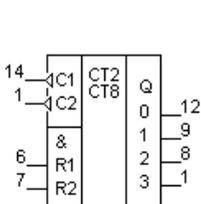
Все рассмотренные выше счетчики являются асинхронными. В них каждый последующий триггер управляется предыдущим. Это приводит к временным задержкам между моментами переключения триггеров, которые на временной диаграмме 9.6 должны переключаться одновременно. Например, с приходом восьмого импульса должны одновременно переключиться все четыре триггера схемы. В действительности же, из-за конечного времени переключения каждого триггера, второй триггер переключится с некоторой задержкой относительно первого, третий – с задержкой относительно второго, и т.д. Несмотря на то, что время переключения современных схем не превышает десятков наносекунд, это может привести к нарушению работоспособности последующих схем. Так, в рассматриваемом случае, счетчик, находящийся в состоянии $Q_3Q_2Q_1Q_0 = 0111$, после восьмого тактового импульса при переключении первого триггера попадает в состояние 0110, потом при переключении второго – в состояние 0100, далее при переключении третьего – в состояние 0000, и только при переключении четвертого - в нужное состояние 1000.

В синхронных счетчиках импульсы синхронизации подаются одновременно на все триггеры. Чтобы с приходом каждого входного импульса не происходило одновременного переключения всех триггеров, схему дополняют многовыходовыми логическими элементами И (рис. 9.9). Каждый триггер схемы теперь может переключиться только при единичных состояниях всех предыдущих триггеров.

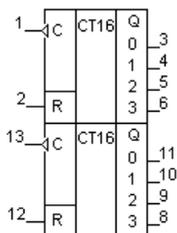
Вход разрешения E предназначен для управления работой счетчика. При $E=0$ работа счетчика прекращается. Сигнал на выходе переполнения P формируется при единичных состояниях всех четырех триггеров. Соединение выхода переполнения одного

На рис. 9.11 переключения триггера D2 происходят после второго, четвертого. Шестого и восьмого импульса. А после десятого триггер не переключается, а устанавливается в нулевое состояние. Триггер D3, также как и D1, работает в счетном режиме. Триггер D4, вход синхронизации которого подключен к выходу D1, при $Q2=Q3=1$ изменяет свое состояние по заднему фронту Q0, а при $Q2 \vee Q3=0$ устанавливается в нулевое состояние. До шестого импульса имеем: $Q2 \vee Q3=0$ и триггер D4 хранит нулевое состояние; к моменту прихода восьмого импульса $Q2=Q3=1$ и после его триггер опрокидывается в единичное состояние; после восьмого снова имеем $Q2=0$, а после десятого триггер D4 вновь возвращается в нулевое состояние.

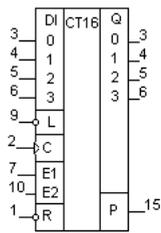
Ряд популярных интегральных счетчиков изображен на рис. 9.12 и 9.13.



1533IE5 (асинхронный счетчик)



555IE19 (два асинхронных счетчика)

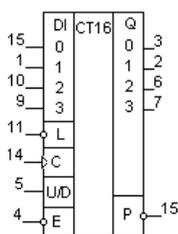


1533IE18 (синхронный счетчик)

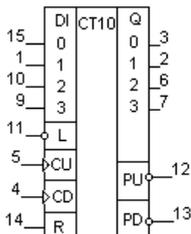
Рис. 9.12. Двоичные счетчики

В схеме счетчика 1533IE5 первый триггер работает как одноразрядный счетчик, а три последующих образуют трехразрядный счетчик с коэффициентом счета 8. Для организации четырехразрядного счетчика с коэффициентом счета 16 необходимо выход Q0 соединить со входом C2. Переключения счетчика происходят по задним фронтам импульсов синхронизации. В режиме счета необходимо выполнять условие: $R1 \cdot R2 = 0$. При $R1 = R2 = 1$ счетчик обнуляется.

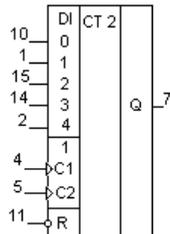
Счетчики микросхемы 555IE19 также считают по задним фронтам импульсов на входах C при R=0. При R=1 происходит



555IE13 (синхронный реверсивный счетчик)



1533IE6 (двоично-десятичный реверсивный счетчик)



555ПЦ1 (управляемый делитель частоты)

Рис. 9.13. Счетчики

Счетчик 1533ИЕ18 при $\#L=0$ по переднему фронту импульса C записывает информацию в триггеры со входов DI . Режим счета устанавливается при условии: $\#R = \#L = E1 = E2 = 1$. Положительный фронт импульса переполнения на выходе P формируется при переходе счетчика в состояние $Q0=Q1=Q2=Q3=1$.

В счетчик 555ИЕ13 также при $\#L = 0$ по переднему фронту импульса C записываются данные со входов DI . Счет разрешается при $\#E = 0$. При $U/D=1$ счетчик является суммирующим, а при $U/D=0$ - вычитающим.

Реверсивный счетчик 1533ИЕ6 имеет отдельные входы суммирования CU и вычитания CD и, соответственно, разные выходы переноса: PU - при суммировании и PD - при вычитании. Предварительная запись данных в счетчик осуществляется с входов DI при $\#L=0$. Счетчик обнуляется при $R = 1$.

Счетчик 555ПЦ1 выполняет функции делителя частоты. Частота выходного сигнала счетчика в 2^N раз меньше входной. Значение N задается в двоичном параллельном коде на входах DI . Входной сигнал подается на любой из входов C , объединенных логикой ИЛИ. При $\#R=0$ счетчик обнуляется и в режиме деления не работает.

10. ВЕНТИЛЬНЫЕ МАТРИЦЫ

Специализированные большие интегральные схемы (БИС), содержащие на одном кристалле сотни тысяч и миллионы логических элементов, позволяют решать задачи любой степени сложности. Однако их разработка и производство требуют значительных затрат времени и материальных ресурсов на всех стадиях производства от проектирования до изготовления. Методы проектирования и изготовления БИС выбираются в зависимости от технико-экономических требований. По способам проектирования и подготовки производства современные БИС условно разделяются на две группы: *заказные* и *полузаказные* микросхемы.

Заказные микросхемы характеризуются наилучшими электрическими параметрами, но требуют индивидуального проектирования и разработки комплекта специализированных фотошаблонов. Длительность цикла разработки заказной БИС от функциональной схемы до готового изделия составляет 20...50 недель. Высокая стоимость разработки заказных микросхем окупается, как правило, только при годовых объемах выпуска более 1 млн. штук.

Полузаказные микросхемы создаются на основе *вентильных матриц* GA (*Gate Array*). В отечественной литературе их также называют *базовыми матричными кристаллами*.

Вентильная матрица GA представляет собой набор, размещенных на кристалле, базовых ячеек, между которыми оставлены свободные зоны для создания межсоединений. Массив базовых ячеек размещается в центральной части GA , а по периферии – периферийные ячейки, из которых могут быть созданы входные и выходные каскады для согласования БИС с другими микросхемами. Из элементов базовых ячеек на кристалле может быть сформирован один логический элемент, а для реализации более сложных функций используется несколько ячеек.

Например, вентильная матрица $K1527XM1$, изображенная на рис. 10.1, на площади $6,35 \times 5,6$ мм содержит матрицу из $12 \times 8 = 96$ базовых ячеек и ещё дополнительную правую колонку, состоящую из восьми левых половин базовых ячеек. Периферийная часть вентильной матрицы содержит 44 периферийных ячейки с контактными площадками и отдельные контакты для подключения источников питания $+5B$ и $+2,5B$.

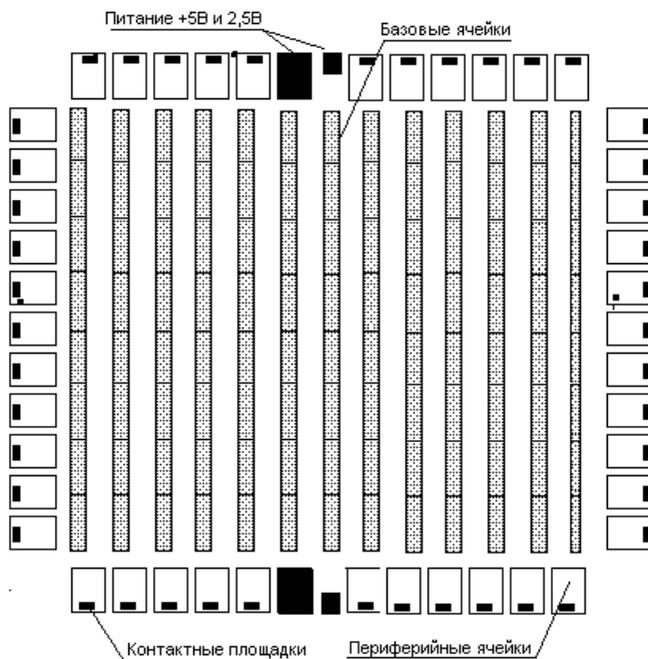


Рис. 10.1. Конструкция микросхемы K1527XM1

Каждая базовая ячейка микросхемы состоит из четырех фрагментов в виде левой NL, OL и правой NR, OR половин. Все фрагменты являются зеркальными отражениями друг друга и состоят из идентичных наборов некоммутированных транзисторов, диодов и резисторов (рис. 10.2).

Каждая периферийная ячейка имеет по два фрагмента (фрагменты 1 и 2), отличающихся друг от друга составом некоммутированных компонентов (рис. 10.3). Фрагменты позволяют строить различные схемы входных и выходных каскадов микросхемы.

Функциональной ячейкой вентильной матрицы является схема, реализуемая путем соединения элементов в пределах одной или нескольких базовых или периферийных ячеек.

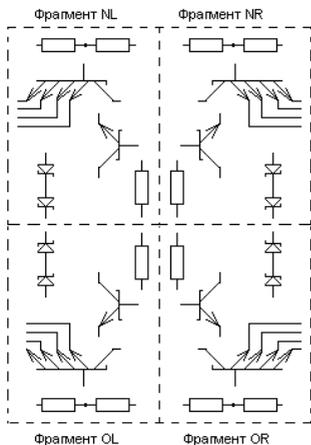


Рис. 10.2. Базовая ячейка микросхемы K1527XM1

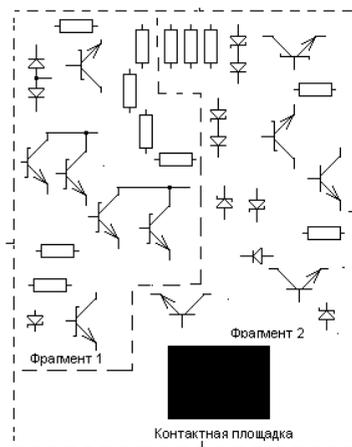


Рис. 10.3. Периферийная ячейка микросхемы K1527XM1

Для каждого кристалла разрабатывается библиотека функциональных ячеек, используемых при проектировании схем. Так, библиотека микросхемы K1527XM1 содержит 32 типа матричных функциональных ячеек (МО1-М32) и 6 типов периферийных функциональных ячеек (ПО1-ПО6). Матричные функциональные ячейки, построенные на основе базовых ячеек в составе микросхемы K1527XM1, реализуют функции:

- МО1...МО8 - простой элемент И-НЕ на базе одного многоэмиттерного транзистора (до четырех входов);
- МО9... М12 - инвертор без многоэмиттерного транзистора,
- М13... М20 - элемент И-НЕ с расширением по И (до восьми входов),
- М21... М28 - элемент И (до четырех входов),
- М29...М32 - элемент И (до восьми входов),

Кроме того, функциональные ячейки могут иметь выходы с открытым коллектором, ограничительные диоды на выходах и различную ориентацию на кристалле.

Периферийные функциональные ячейки в микросхеме K1527XM1 реализуют функции:

- ПО1 – выходной буфер с открытым коллектором,
- ПО2 – выходной TTL-буфер с резистивной нагрузкой,
- ПО3 – выходной TTL-буфер;
- ПО4 – выходной TTL-буфер с тремя состояниями,
- ПО5 – входной буфер с p-n-p транзистором,
- ПО6 – входной буфер с открытым коллектором.

Библиотека функциональных ячеек размещается на базовых ячейках с помощью металлизированных соединений элементов, выполненных в соответствии с электрическими схемами функциональных ячеек. Для проектирования металлизированных соединений в вентиляльную матрицу вводится сетка проектирования, имеющая фиксированный шаг по вертикали и горизонтали. При трассировке соединений используются вертикальные и горизонтальные каналы между колонками базовых ячеек. Например, у вентиляльной матрицы K1527XM1 шаг сетки проектирования равен 16 мкм, а при трассировке допускается использовать 165 вертикальных каналов и 115 горизонтальных.

Проектирование металлизированных соединений осуществляется с помощью специализированной *системы автоматизированного проектирования (САПР)*, разрабатываемой изготовителем вентиляльной матрицы. Используемые при проектировании САПР, как правило, ориентированы на узкий круг выпускаемых изделий. Они включают в себя синтез функциональной электрической схемы на основе базовых функциональных ячеек, логическое моделирование, синтез тестов контроля, разработку топологии межсоединений, расчет электрических параметров и моделирование работы схемы с учетом реальной топологии межсоединений, изготовление фотошаблонов, изготовление и испытания опытных образцов.

При изготовлении кристалла используются все основные технологии производства цифровых микросхем: транзисторно-транзисторная логика с диодами Шоттки, эмиттерно-связанная логика, n-МОП и CMOS – логика. Кроме того, разработаны сверхскоростные GA на *арсенид-галлиевых транзисторах* со структурой *металл-полупроводник (МЕП – транзисторы)*.

Сравнительные данные некоторых отечественных вентиляльных матриц приведены в таблице 10.1.

Таблица 10.1.

Отечественные вентиляльные матрицы

Тип	Технология	Размер кристалла (мм)	Базовых ячеек	Контактов	Задержка (нс)	Частота МГц
1521XM1	ЭСЛ	4,4x3,3	36	56	1,2	
1520XM2	ЭСЛ	4,4x3,3	72	108		
1572XM1	ЭСЛ		18, 36, 72	65	1,9	
1527XM1	ТТЛШ	6,35x5,6	96	47	6	
1548XM1	ТТЛШ	7,5x7,3	247	89	5	
1598XM1	ТТЛШ		600	64	7	
1801ВП1	n-МОП	4,2x4,2	520	43	5	8
1806ВП1	CMOS	5,2x5,2	416	42		4
1515XM1	CMOS	6,35x7,15	1012	64	5	10
K6501XM1	МЕП	4,5x3,4	25	56	0,15	1000

В настоящее время зарубежные изготовители предлагают потребителям целые ряды базовых матричных кристаллов с различным числом логических элементов. Изделия каждого ряда могут иметь различные значения быстродействия и потребляемой мощности, изготавливаться по различным проектным нормам, иметь различное число выводов. В таблице 10.2 приведено несколько примеров таких рядов.

Таблица 10.2.

Ряды базовых матричных кристаллов

Фирма	Минимальный размер элемента (мкм)	Число типов БМК в ряду	Число логических элементов на кристалле	Время задержки элемента (нс)	Потребляемая мощность (мВт/элемент)
LSI Logic	3	7	880...6000	2,2	
	2	23	554...10013	1,4	
	1,5	18	980...129042	1,1	
Ferranti	3	6	130...2000	2,5	0,3
		11	130...4000	7,5	0,1
		7	130...2000	15	0,03
		13	400...10000	15	0,15
		8	400...4000	3	0,07
		8	400...4000	10	0,02

Такое разнообразие базовых матричных кристаллов позволяет потребителю выбрать кристалл, наилучшим образом соответствующий проектируемой схеме.

11. ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ ИНТЕГРАЛЬНЫЕ СХЕМЫ

Микросхемы PLD (*Programmable Logic Devices*) - это большие интегральные схемы, позволяющие программно сконструировать в одном корпусе электронную схему, содержащую сотни интегральных схем простейшей логики. В отечественной литературе они называются программируемыми логическими интегральными схемами (ПЛИС). Микросхемы PLD представляют собой матрицы логических ячеек, соединенных между собой логическими ключами. Содержащаяся в микросхемах память, будучи заполненной определенным образом, воздействует на логические ячейки и соединяющие их ключи, позволяет синтезировать различные логические схемы как комбинационного, так и последовательного типа.

Общепринятой оценкой логической ёмкости PLD является число эквивалентных вентилях типа 2И-НЕ. Эта оценка весьма условна, поскольку микросхемы PLD не содержат вентилях 2И-НЕ в чистом виде, однако для проведения сравнительного анализа различных архитектур она вполне пригодна. Основным критерием классификации современных микросхем PLD является наличие, вид и способы коммутации элементов логических матриц. По этому признаку выделяют несколько классов PLD.

Микросхемы FPLA

Микросхемы FPLA (*Field Programmable Logic Array*) или программируемые логические матрицы (ПЛМ) — наиболее традиционный тип микросхем, имеющих программируемые матрицы И и ИЛИ. Они являются универсальными логическими схемами, предназначенными для реализации систем логических функций, заданных в дизъюнктивной нормальной форме. Схема FPLA состоит из двух программируемых матриц (матрицы логических элементов И и матрицы логических элементов ИЛИ) и вспомогательных логических схем, обеспечивающих подключение и программирование микросхемы. Например, отечественная микросхема К556РТ2 (рис. 11.1) содержит программируемую матрицу И, программируемую матрицу ИЛИ, входные усилители,

выходные усилители, программирующий дешифратор и программирующие адресные формирователи.

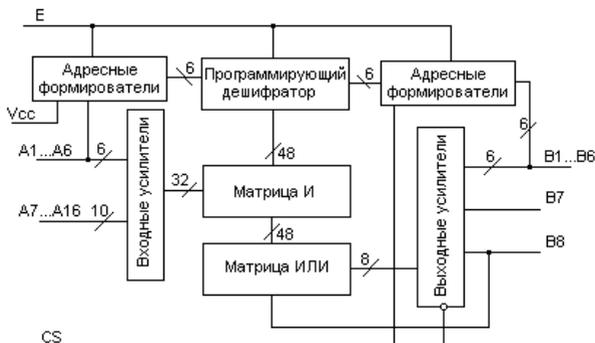


Рис. 11.1. Структура микросхемы K556PT2

Входные усилители в K556PT2 формируют и передают на матрицу И прямые и инверсные значения входных переменных A1...A16. В состав матрицы И входят 48 конъюнкторов, подсоединенных при помощи плавких нихромовых переключателей к выходам усилителей. В матрице И реализуются конъюнкции входных переменных, причем в зависимости от состояния пережигаемых переключателей каждая переменная может войти в конъюнкцию прямым или инверсным значением или не войти вовсе. Выходные сигналы матрицы И (P1...P48) поступают на матрицу ИЛИ, реализующую дизъюнкцию заданных конъюнкций. Матрицу ИЛИ образует 8 дизъюнкторов, каждый из которых через плавкие переключатели связан со всеми выходами P1...P48 матрицы И. Выходы S1...S8 матрицы ИЛИ подсоединены к выходным усилителям. Последние также являются программируемыми и позволяют программировать активный уровень выходного сигнала (низкий или высокий). Программирующий дешифратор вместе с адресными формирователями работает только в режимах программирования матриц И и ИЛИ. Для формирования требуемой конъюнкции программирующий дешифратор подключает к источнику программирующего тока пережигаемые переключатели матрицы И. Для формирования необходимой дизъюнкции он же подключает к источнику программирующего тока пережигаемые переключатели матрицы ИЛИ. Первый адресный формирователь используется при выборе переключателей в матрице ИЛИ, второй – в матрице ИЛИ.

Режимы работы микросхемы задаются сигналами на управляющих входах. Для программирования микросхем также используются двунаправленные выходы. Например, рассмотренная микросхема K556PT2 (рис. 11.1 и 11.2) в режиме считывания работает при напряжении питания $V_{CC} = 5В$. На вход выбора микросхемы CS подается низкий уровень сигнала ($CS = 0$), соответствующий стандартному значению логического нуля для TTL-логики. Входные переменные A1...A16 в этом случае через блок входных усилителей поступают на матрицу И, созданные матрицей И конъюнкции передаются на матрицу ИЛИ, сигналы с которой через выходные усилители проходят на выходы V1...V8 микросхемы.

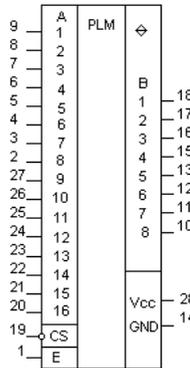


Рис. 11.2. Микросхема K556PT2

В режиме программирования матрицы И при напряжении питания $V_{cc} = 5В$ на вход выбора микросхемы CS подается напряжение программирования 10 В, на выходы $B1...B6$ – двоичный код адреса конъюнкции ($P1...P48$). На адресный вход A , соответствующий пережигаемой перемычке, подается напряжение высокого уровня в пределах 3...4,5 В, если нужно пережечь перемычку, подсоединенную к инверсному выходу входного усилителя, или напряжение низкого уровня 0...0,5 В, если пережигается перемычка, подсоединенная к прямому выходу входного усилителя. На остальные адресные входы из множества $A1...A16$ подается напряжение 10В.

За каждый цикл программирования пережигается только одна перемычка. Импульс программирующего тока формируется при подаче на программирующий вход E напряжения не менее 17 В.

Программирование матрицы ИЛИ осуществляется при повышенном напряжении питания $V_{cc} = 8,75 В$. При этом на входы $A1...A6$ подается код, соответствующий номеру логического произведения $P1...P48$, которое надо исключить из функции, на вход CS – напряжение 3...4,5 В, на выход соответствующий функции, из которой исключается выбранная конъюнкция, подается напряжение 10 В. Импульс программирующего тока формируется при подаче на программирующий вход E напряжения 17В, а на вход CS – напряжения 10 В. За каждый цикл программирования матрицы ИЛИ также пережигается только одна перемычка.

Программирование выходных усилителей производится подачей на соответствующий выход $B1...B8$ напряжения программирования 17 В при напряжении на входе CS равном 3...4,5 В и напряжении питания $V_{cc} = 0$. При пережигании перемычки в выходном усилителе он становится инвертирующим.

Общим недостатком микросхем FPLA, ограничивающим их развитие, является слабое использование ресурсов программируемой матрицы “ИЛИ”.

Микросхемы PAL

Микросхемы *PAL (Programmable Array Logic)*, или программируемая матричная логика (ПМЛ), имеют программируемую матрицу “И”, фиксированную матрицу “ИЛИ”, макро ячейки (*macro cells*) и цепи обратной связи, связывающие выходы схемы со входами матрицы И. (рис. 11.3). Макро ячейки могут включать в себя шинные формирователи с тремя выходными состояниями, триггеры, мультиплексоры.

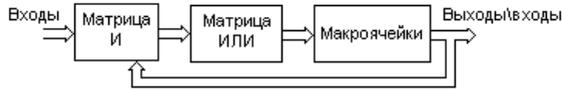


Рис. 11.3. Структура микросхемы PAL

Например, микросхема КМ1556ХП4 (рис. 11.4) содержит программируемую матрицу И, матрицу ИЛИ, входные усилители, макроячейки в виде регистра на D-триггерах, адресный формирователь обратной связи, блок разрешения выборки, выходной усилитель, программирующий дешифратор.

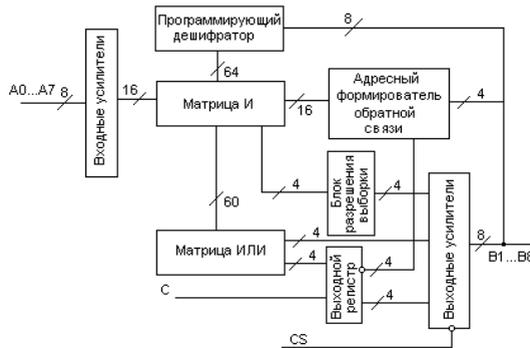


Рис. 11.4. Структурная схема КМ1556ХП4

Матрица И микросхемы КМ1556ХП4 содержит 60 элементов И, связанных через плавкие перемычки с 16 входными переменными и их инверсиями через входные усилители и адресные формирователи обратной связи. Шесть входных переменных из 16 поступают с выходного регистра. В матрице И образуются конъюнкции входных переменных, причем каждая входная переменная может входить в конъюнкции прямым, инверсным значением или не входить вовсе.

Выходные сигналы с шин матрицы И поступают на матрицу ИЛИ. Матрицу ИЛИ образуют 8 элементов ИЛИ, каждый из которых может быть связан с одной из восьми групп элементов И матрицы И.

Входные усилители формируют множество прямых и инверсных значений входных переменных A0...A7. Адресные формирователи обратной связи формируют аналогичное множество прямых и инверсных сигналов обратной связи.

Выходной регистр образуют четыре D-триггера, тактируемые сигналом синхронизации С. С инверсных выходов D-триггеров сигналы поступают на адресные формирователи обратной связи.

Выходной каскад включает в себя восемь усилителей с тремя состояниями, четыре из которых управляются сигналом CS, а четыре других – из матрицы И через блок разрешения выборки.

Программирующий дешифратор выбирает при программировании пережигаемые перемычки матрицы И. Для управления дешифратором используются из восьми выходных контактов микросхемы (рис. 11.5) используется только шесть (B1...B6).

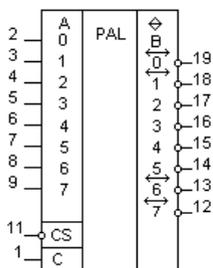


Рис.11.5. Микросхема KM1556ХП4

В рабочем режиме входные сигналы подаются на входы A0...A7 и двунаправленные выходы B0, B1, B6, B7. Сигналы через входные усилители и адресные формирователи обратной связи попадают на матрицу И. Одновременно на матрицу И через формирователь обратной связи поступают сигналы с выходного регистра. При подаче тактового импульса С на выходах схемы формируется логическая функция от входных сигналов и внутренних состояний регистра. Разрешение выборки при считывании осуществляется при низком уровне сигнала CS (для выходов B2...B5) или непосредственно из матрицы И (для выходов B1, B2, B6, B7).

Программирование микросхем программируемой матричной логики осуществляется аналогично программированию FPLA.

За счет различных связей между элементами схемы PAL позволяют реализовать довольно сложные устройства. На PAL без триггеров возможно построение только комбинационных схем, таких как дешифраторы, мультиплексоры, сумматоры. PAL с триггерами, позволяют строить также и последовательностные схемы: счетчики, управляющие автоматы, контроллеры. Проектирование схем на основе PAL возможно только на основе специализированных систем автоматизированного проектирования (САПР), поставляемых разработчиком микросхемы. Для программирования используют программаторы, поставляемые вместе с пакетом САПР.

При изготовлении PAL используются в основном TTL и CMOS технологии. Микросхемы, изготовленные по технологии CMOS с ультрафиолетовым стиранием, допускают неоднократное перепрограммирование. Сравнительные данные ряда зарубежных и отечественных микросхем PAL приведены в таблице 12.1.

Таблица 12.1.

Основные характеристики микросхем PAL

Тип PAL (фирма разработчик)	Отечественный аналог	Степень интеграции (вентилей)	Число выводов	Число триггеров	Быстродействие

Тип PAL (фирма разработчик)	Отечественный аналог	Степень интеграции (вентилей)	Число выводов	Число триггеров	Быстродействие
Технология TTLШ					
PLS101 (Signetics)	556PT1	300	28	0	60 нс
PLS100 (Signetics)	556PT2	300	28	0	60 нс
PLS105 (Signetics)	556PT2	300	28	14	60 нс
PAL16L4 (AMD)	1556XJL8	300	20	0	25...45 нс
PAL16R4 (AMD)	1556XП4	300	20	4	25...45 нс
PAL16R4 (AMD)	1556XП4	300	20	4	25...45 нс
PAL16R6 (AMD)	1556XП6	300	20	6	25...45 нс
PAL16R8 (AMD)	1556XП6	300	20	8	25...45 нс
Технология УФ CMOS					
85C060 (Intel)	1558ЯТ1	600	24	16	40...100 МГц
85C090 (Intel)	573ХМ330	900	40	24	50...83 МГц
85C22V10 (Intel)		600	24	10	66...100 МГц
85C220 (Intel)		300	20	8	100...120 МГц
85C224 (Intel)		400	24	8	100...120 МГц

Вышеперечисленные архитектуры программируемых логических интегральных схем содержат небольшое число ячеек, к настоящему времени морально устарели. Они применяются для реализации относительно простых устройств, для которых не существует готовых схем средней степени интеграции.

Микросхемы CPLD

Микросхемы *CPLD (Complex Programmable Logic Devices)*, или программируемые коммутируемые матричные блоки (ПКМБ), имеют архитектуру, весьма удобную для реализации цифровых автоматов. Они имеют несколько матричных логических блоков со структурой PAL, объединённых коммутационной матрицей. CPLD, как правило, имеют высокую степень интеграции (до 10000 эквивалентных вентилей, до 256 макроячеек). К этому классу относятся ПЛИС семейства MAX5000 и MAX7000 фирмы ALTERA, схемы XC7000 и XC9500 фирмы XILINX, а также большое число микросхем других производителей (Atmel, Vantis, Lucent и др.).

Микросхемы FPGA

Микросхемы *FPGA (Field Programmable Gate Array)*, как и вентильные матрицы типа GA, состоят из множества логических блоков и блоков ввода-вывода, связанных между собой посредством коммутационных блоков. Но, в отличие от вентильных матриц, логические блоки, блоки ввода-вывода и коммутационные поля конфигурируются при загрузке в микросхему битовой последовательности, полученной в результате разработки схемы. В отечественной литературе такие схемы получили название программируемые вентильные матрицы.

Логические блоки таких микросхем состоят из одного или нескольких относительно простых логических элементов, в основе которых лежит таблица перекодировки (*Look-up table* — LUT), программируемые мультиплексоры PMS, D-триггеры, а также цепи

управления (рис. 11.6). Логический блок может выполнять любую логическую функцию в соответствии с заданной битовой последовательностью. Изменять выполняемую функцию можно неограниченное количество раз путем загрузки в микросхему другой битовой последовательности.

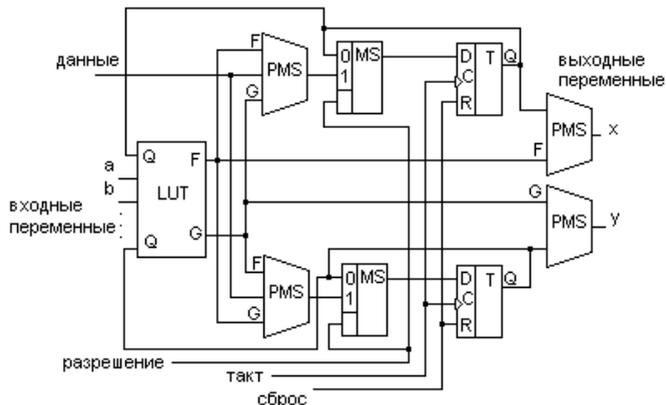


Рис. 11.6. Пример схемы логического блока микросхем FPGA

Количество логических блоков у современных микросхем FPGA ёмкостью до 1 млн. вентилях достигает нескольких десятков тысяч. Они содержат значительное число триггеров, а также могут иметь встроенные реконфигурируемые модули памяти (МП), *embedded array block* — EAB. К FPGA классу относятся микросхемы XC2000, XC3000, XC4000, Spartan, Virtex фирмы XILINX; ACT1, ACT2 фирмы ACTEL, а также семейства FLEX8000 фирмы ALTERA, некоторые микросхемы фирм Atmel и Vantis.

12. МИКРОСХЕМЫ ПАМЯТИ

Микросхемы памяти предназначены для хранения больших объемов цифровой информации. Объем записываемой и хранимой в памяти информации обычно измеряется в байтах (8 бит), килобайтах ($2^{10} = 1.024$ байт), мегабайтах ($2^{20} = 1.048.576$ байт) и гигабайтах ($2^{30} = 1.073.741.824$ байт).

Микросхемы памяти находят самое различное применение в вычислительных устройствах и по своему функциональному назначению делятся на две большие группы:

Оперативные запоминающие устройства (ОЗУ) обеспечивают возможность записи и чтения любой ячейки памяти в процессе работы. Микросхемы ОЗУ энергозависимы. Они способны хранить информацию только при включенном питании. При отключении источника питания, даже кратковременном, вся информация в ОЗУ теряется. Термин ОЗУ примерно соответствует английскому *RAM (Random Access Memory)* – память с произвольным доступом.

Постоянные запоминающие устройства (ПЗУ) предназначены для хранения неизменяемых данных. Они энергонезависимы, т.е. способны хранить информацию при

отсутствии питающего напряжения Термин ПЗУ соответствует английскому *ROM (Read Only Memory)* - память только для чтения.

Оперативные запоминающие устройства

Оперативные запоминающие устройства строятся на различной элементной базе. В зависимости от используемых ячеек памяти они разделяются на две группы: *статические (Static RAM)* и *динамические (Dynamic RAM)*. Ячейки памяти статических ОЗУ представляют собой триггерные ячейки, позволяющие хранить информацию при включенном питании в течении неограниченного времени. В динамических ОЗУ в качестве ячеек памяти используются конденсаторы, способные в течении короткого времени хранить электрический заряд. Использование таких элементов требует периодического обновления (регенерации) информации.

Микросхемы SRAM

В *статическом ОЗУ (SRAM)* ячейки памяти объединяются в матрицу. Матрица состоит из k строк, каждая из которых объединяет n ячеек памяти. Количество строк и столбцов определяет емкость ОЗУ $N = k \cdot n$ (бит). Для доступа к ячейкам памяти в схеме ОЗУ предусматриваются: дешифратор строк и дешифратор столбцов (рис. 12.1).

Общее множество входов дешифратора строк и дешифратора столбцов $A_0 A_1 A_2 A_3 \dots A_m$ образует двоичный адрес ячейки памяти. Причем разрядность адреса однозначно связана с емкостью ОЗУ известным соотношением: $2^{m+1} = N$. Работой микросхемы управляет устройство ввода/вывода, на которое поступают данные для записи в выбранную ячейку памяти DI (*Data Input*), сигнал выбора микросхемы CS (*Chip Select*) и сигнал разрешения записи WE (*Write Enable*), разрешения выхода OE (*Output Enable*). Данные из выбранной ячейки памяти поступают на выход DO (*Data Output*).

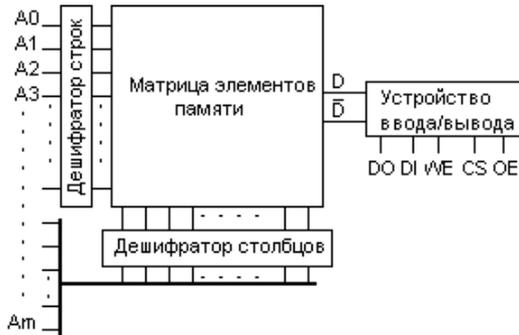


Рис. 12.1. Структура микросхемы SRAM

Для записи в микросхему одного бита данных необходимо подать эти данные на вход DI , подать на адресные входы код адреса, нулевой разрешающий сигнал \overline{CS} и сигнал $WE = 1$, соответствующий режиму записи. При указанных сигналах выбирается одна строка и один столбец матрицы, соответствующие коду адреса. Входные данные в парафазном виде (по двум линиям D и \overline{D}) через устройство ввода/вывода поступают на общие шины

матрицы, связанные со всеми ячейками памяти, и записываются в ячейку, лежащую на пересечении выбранной строки с выбранным столбцом.

Каждый элемент памяти статического ОЗУ представляет собой триггер, который может быть реализован на биполярных или полевых транзисторах. Триггер на рис. 12.2 собран по схеме CMOS-логики.

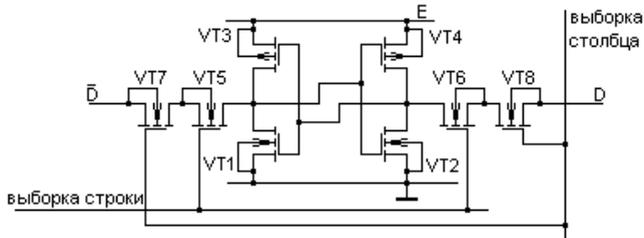


Рис. 12.2. Статический элемент памяти CMOS-логики.

Здесь транзисторы $VT1$ и $VT2$ с каналом типа n , а транзисторы $VT3$, $VT4$ – с каналом p -типа. Триггер имеет два совмещенных входа-выхода D и \bar{D} . Транзисторы $VT5$, $VT6$, $VT7$, $VT8$ соединяют триггер с разрядными шинами, по которым в парафазном виде подводится к триггеру при записи (или отводится от него при считывании) информация: Транзисторы $VT5$, $VT6$ затворами подсоединены к строке, $VT7$, $VT8$ – к столбцу. При выборе ячейки в режиме чтения данные D и \bar{D} поступают на общие шины, проходящие через все ячейки матрицы, считываются усилителем и через устройство ввода/вывода поступают на выход микросхемы. В режиме записи данные с входа микросхемы через устройство ввода/вывода поступают на шины и через открытые транзисторы $VT5$, $VT6$, $VT7$, $VT8$ попадают на триггер и устанавливают его в соответствующее состояние.

Рассмотренная структурная схема ОЗУ имеет одnorазрядную организацию. Этот тип микросхем является преобладающим, но не единственным. Вместе с ним широко применяются микросхемы статических ОЗУ с многоразрядной (словарной) организацией. В последних матрица элементов памяти разбивается на отдельные секции, а запись и чтение информации производится одновременно из всех секций.

Современные микросхемы SRAM имеют несколько разновидностей, имеющих различные способы выборки данных. Под термином SRAM обычно подразумевается *асинхронная статическая память (Asynchronous SRAM)*, в которой выборка данных производится при нулевых сигналах CS (*Chip Select*), OE (*Output Enable*), запись – при нулевом сигнале WE (*Write Enable*). Задержка времени, между моментом установки адреса и моментом появления данных на выходе в режиме считывания (*время доступа*) у современных асинхронных микросхем SRAM составляет 12...20 нс.

. На рис. 12.3 изображен ряд отечественных микросхем SRAM с различной организацией данных.

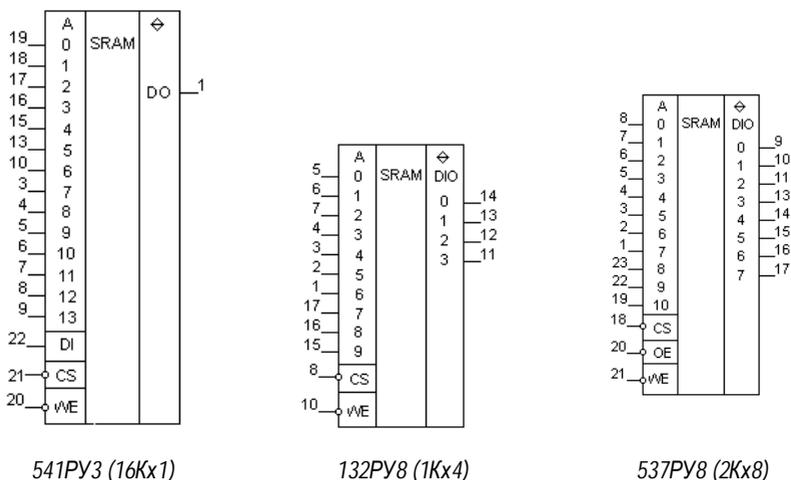


Рис. 12.3. Микросхемы SRAM

Микросхема 541PY3 имеет одноразрядную организацию, микросхема 132PY8 – четырехразрядную, а 537PY8 – восьмиразрядную. Все микросхемы имеют выходы с тремя состояниями. Третье (высокоимпедансное) состояние присутствует на выходах при $\overline{CS} = 1$, т.е. когда микросхема не выбрана. В микросхеме это состояние появляется также при отсутствии сигнала разрешения (при $\overline{OE} = 1$).

Современные микросхемы асинхронной SRAM обычно исполняются восьмиразрядными и имеют объем 256, 512 или 1024 Кбайт.

Синхронная статическая память (*Sync Burst SRAM*) предназначена для выполнения пакетных (*burst*) операций обмена, в которых за один цикл обращения к памяти производится последовательное считывание блока данных из четырех соседних ячеек. Для этого в микросхему введен внутренний двухбитный счетчик адреса. В дополнение к сигналам, используемым асинхронной памятью (адрес, данные, CS, OE, WE), в ней предусмотрен сигнал синхронизации *CLK (Clock)*. В пакетном режиме передача первого слова данных осуществляется обычно за 2 - 3 такта, а передача каждого из трех остальных - за один такт.

Микросхемы синхронной памяти обычно имеют разрядность 16 или 32 бит. Время доступа у них колеблется в пределах 8,5 – 12 нс.

Микросхемы DRAM

В *динамических ОЗУ (DRAM)* используются наиболее простые элементы памяти – конденсаторы, объединяемые также как и в статическом ОЗУ, в матрицу. Адрес элемента памяти задается строкой и столбцом матрицы. При записи логической единицы выбранный конденсатор заряжается, а при записи нуля – разряжается. Схема считывания данных разряжает через себя выбранный конденсатор и, если заряд был ненулевым, формирует единичный сигнал и, одновременно, подзаряжает конденсатор до прежнего уровня. На рис. 10.4 показан один из вариантов схемы считывания данных из столбца матрицы.

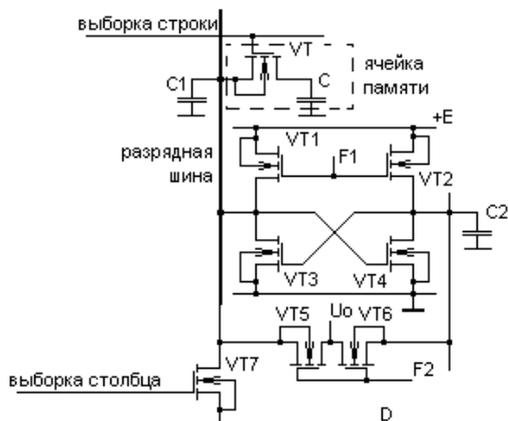


Рис. 12.4. Схема считывания данных из ячейки DRAM

Элемент памяти состоит из конденсатора C и транзистора VT . Транзистор выполняет функции ключа: он открывается по сигналу выборки строки и соединяет конденсатор C с разрядной шиной. Предварительно в паузах между обращениями к строкам матрицы емкости шин $C1$ и $C2$ заряжаются от источника напряжения U_0 через открытые ключевые транзисторы $VT5$ и $VT6$. При обращении к строкам эти транзисторы закрываются. Запоминающий конденсатор C выбранного элемента памяти при подключении к разрядной шине изменяет её потенциал. Это изменение незначительно, т.к. емкость запоминающего конденсатора много меньше емкости шины $C1$. Поэтому при считывании информации используется дифференциальный усилитель триггерного типа на транзисторах $VT1...VT4$. Поскольку напряжение на конденсаторе $C2$ отличается от напряжения на разрядной шине, то триггер опрокидывается и на разрядной шине формируется уровень сигнала, соответствующий считываемой с конденсатора информации. При этом происходит восстановление заряда конденсатора, т.е. регенерация элемента памяти. Эта операция одновременно происходит во всех элементах выбранной строки. При выборе столбца матрицы информация поступает на выход микросхемы.

На рис. 12.5 изображена структурная схема микросхемы DRAM. В отличие от ОЗУ статического типа схема содержит два адресных регистра с объединенными входами. Ячейка памяти в микросхеме DRAM выбирается за два этапа. Сначала на адресные входы подается код адреса строки и импульс RAS (Row Address Select), производящий запись кода в регистр адреса строки. Далее на те же адресные входы подается код адреса столбца и импульс CAS (Column Address Select), производящий запись кода в регистр адреса столбца. Ячейка памяти выбирается на пересечении выбранной строки и выбранного столбца. Устройство ввода вывода управляет процессами чтения и записи информации.

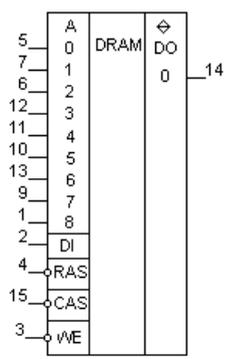


Рис. 12.5. Структурная схема микросхемы DRAM

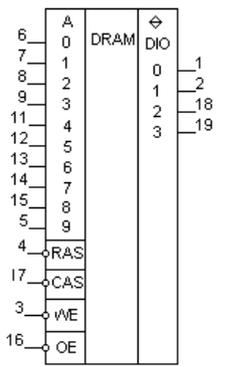
Регенерация памяти (*Memory Refresh*) происходит одновременно по всей строке матрицы при обращении к любой из её ячеек. Максимальный период обращения к каждой строке T_{ff} (*refresh time*) для гарантированного сохранения информации у современной памяти лежит в пределах 8...64 мс.

Современные микросхемы DRAM имеют емкость 1...256 бит, время доступа 45...250 нс и обычно организованы по 1, 4, 8, 16, 18, 32 или 36 бит в корпусе. На рис. 12.6 изображен ряд различных микросхем DRAM.

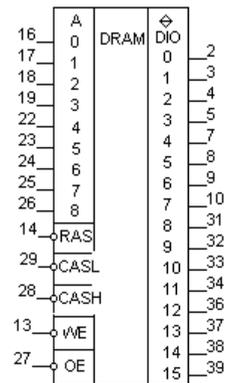
Шестнадцатиразрядная микросхема 4160 состоит из двух половин по 8 бит, имеющих раздельные сигналы CASL и CASH, обеспечивающих возможность побайтного обращения.



565PY7
256Kx1



MCM 4400
1Mx4



MCM 4160
256Kx16

Рис. 12.6. Микросхемы DRAM

При последовательном обращении к ячейкам ОЗУ, принадлежащем одной строке матрицы, адрес строки выставляется на шине только один раз и сигнал RAS удерживается на низком уровне на время всех последующих циклов обращения, которые могут быть

циклами записи или чтения (рис. 12.7). Такой режим обращения к памяти называется режимом быстрого страничного обмена *FPM (Fast Page Mode)*. Под «страницей» в данном случае подразумевается строка, а состояние микросхемы с низким уровнем сигнала RAS называется «открытой страницей».

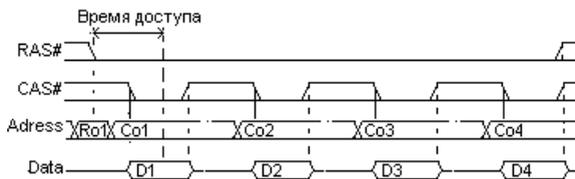


Рис. 12.1. Диаграммы FPM DRAM

Информация на выходе FPM DRAM появляется с некоторой задержкой относительно спада импульса CAS и держится во время его низкого уровня.

Микросхемы динамической памяти *EDO DRAM (Extended Data Out DRAM)*, появившиеся на смену стандартной DRAM, содержат регистр выходных данных. Регистр записывает данные при низком уровне сигнала CAS, а по его подъему переходит в режим хранения. За счет этого данные сохраняются на выходе при исчезновении сигнала CAS и на их считывание внешними схемами отводится больший интервал времени.

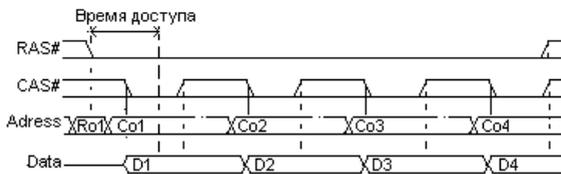


Рис. 12.8. Диаграммы EDO DRAM

В микросхемах *BEDO DRAM (Burst EDO DRAM)* кроме выходного регистра содержится ещё и внутренний 2-битный счетчик адреса столбца, предназначенный для организации пакетного цикла. Адрес столбца записывается в регистр столбца в начале пакетного цикла, а последующие три адреса формируются счетчиком при поступлении импульсов CAS. В результате три последующие ячейки памяти опрашиваются последовательно за меньший промежуток времени.

Синхронная DRAM (*SDRAM - Synchronous DRAM*) отличается от обычной наличием сигнала синхронизации, в качестве которого обычно используется сигнал тактовой частоты системной платы. Это позволяет организовать внутри микросхемы высокопроизводительный конвейер на основе ячеек динамической памяти с обычным временем доступа 50...70 нс. Синхронный интерфейс обеспечивает трехкратный выигрыш по производительности по сравнению с обычными DRAM. Микросхемы SDRAM являются устройствами с программируемыми параметрами, со своим набором команд и внутренней организацией чередования банков. Кроме команд записи и чтения с программируемыми параметрами пакетного цикла имеются команды автоматической регенерации и перехода в режим хранения данных с пониженным энергопотреблением.

Для повышения производительности динамической памяти в новых разработках она дополняется статической *кэш-памятью*, построенной на том же кристалле. Быстродействующий кэш является буфером между DRAM и внешними устройствами, осуществляющими запись и чтение памяти. Кэш хранит копии блоков данных тех областей DRAM, к которым проходили последние обращения. Эта архитектура реализована в микросхемах *CDRAM (Cached DRAM)*. Например, микросхемы CDRAM фирм Mitsubishi и Samsung емкостью 4...16 Мбит имеют 16-кбайтный кэш статической памяти со 128-битной внутренней шиной данных.

Постоянные запоминающие устройства

Постоянная память (ROM) хранит записанные данные и при отсутствии питающего напряжения. Основным режимом такой памяти является считывание данных, но способы занесения данных (программирования) микросхем могут быть самые разные. В зависимости от способа программирования все ПЗУ разделяются на несколько групп:

- масочные ПЗУ (*mask ROM*) – микросхемы, программируемые на стадии изготовления с помощью фотошаблонов;
- однократно программируемые ПЗУ (*OTP ROM – One Time Programmable ROM*) – микросхемы, программируемые пользователем;
- репрограммируемые ПЗУ (*EPROM - Erasable Programmable ROM*) – микросхемы допускающие стирание информации и повторное программирование..

В зависимости от способа стирания EPROM различают:

- микросхемы, стираемые ультрафиолетовым облучением *EPROM*;
- электрически стираемые микросхемы (*EEPROM - Electrically EPROM*).

К последнему типу микросхем относится также *флэш-память (Flash Memory)*, с особой технологией построения запоминающих ячеек.

Структура постоянного запоминающего устройства аналогична структуре микросхем SRAM. Элементы памяти также собираются в матрицу, позволяющую путем выбора строки и столбца обратиться к любому элементу. Для изготовления микросхем используют как биполярные так и МОП-технологии, но принцип построения остается неизменным. Элементы памяти в ROM представляют собой просто диодную или транзисторную переключку между строкой и столбцом. На рис. 12.9 показано несколько различных вариантов запоминающих элементов mask ROM.

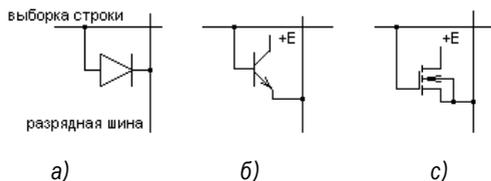


Рис. 12.9. Элементы памяти ROM

а) диод б) биполярный транзистор в) МОП-транзистор

В любой из схем рис. 12.9 при выборке строки на разрядную шину попадает положительное напряжение, соответствующее уровню логической единицы. Если на пересечении строки и разрядной шины элемент памяти отсутствует, то при выборке строки положительное напряжение на разрядную шину не попадает.

На рис. 12.8 изображено простейшее диодное ПЗУ, хранящее четыре четырехбитных слова.

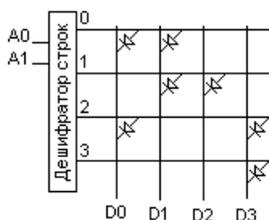


Рис. 12.10. Схема памяти ROM на диодах

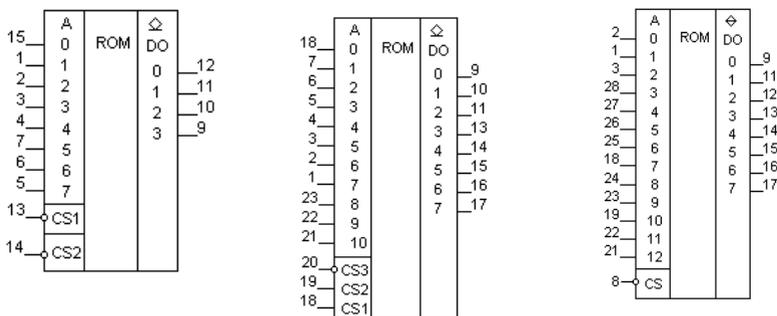
При обращении к дешифратору по линиям A0, A1 выбирается одна из четырех строк матрицы и на выходы схемы D0...D3 поступает занесенная в неё при изготовлении информация. На разрядных шинах (столбцы матрицы) с установленными в выбранной строке диодами появляются единичные сигналы, а на линиях столбцов без диодов – нулевые. Таблица 12.1, связывающая все возможные адресные комбинации и записанные ПЗУ данные (так называемая *таблица прошивки* ПЗУ), соответствует схеме на рис. 12.8.

Таблица 12.1.
Таблица прошивки ROM

Адрес		Данные			
A0	A1	D0	D1	D2	D3
0	0	1	1	0	0
0	1	0	1	1	0
1	0	1	0	0	1
1	1	0	0	0	1

Микросхемы mask ROM

Микросхемы mask ROM имеют стандартные прошивки. Обычно это коды букв различных алфавитов, таблицы математических функций, стандартные коды обмена информации, коды простейших алгоритмических языков (ассемблер, бейсик и др.). Некоторые типы масочных ПЗУ могут быть запрограммированы по заказу пользователя. Несколько микросхем со стандартной прошивкой показано на рис. 12.11.



*155PE21..24 (256x4)
коды букв русского и
латинского алфавитов,
арифметических знаков, цифр*

*555PE4 (2Kx8)
восьмиразрядный код
обмена информацией
КОИ-8*

*568PE2 (8Kx8)
международный телеграфный
код, функции синуса, коды букв
русского и латинского
алфавитов*

Рис. 12.11. Микросхемы mask ROM

Микросхемы масочных ПЗУ могут иметь выходы с открытым коллектором или с тремя выходными состояниями. Сигналы выбора микросхемы CS при определенном их сочетании разрешают считывание информации.

Микросхемы OTP ROM

Однократно программируемые ПЗУ (OTP ROM) по принципу построения и функционирования аналогичны масочным, но в дополнение имеют режим программирования. Операция программирования заключается в избирательном разрушении (пережигании) части плавких перемычек, включенных в элементы памяти. Плавкие перемычки представляют собой небольшой участок металлизации, который расплавляется при программировании импульсами тока. На рис. 12.12 показано включение плавкой перемычки в элемент памяти OTP ROM, построенный по биполярной технологии.



Рис. 12.12. Элемент OTP ROM

Для пережигания перемычки необходимо пропустить через транзистор серию импульсов тока 20...50 мА по специальной методике. Причем каждый тип OTP ROM требует установки своих параметров при программировании. Программирование выполняется сразу со всей матрицей микросхемы по заданной таблице прошивки с помощью специальных приборов – *программаторов*. Методика программирования предусматривает подачу 40...100 импульсов программирования, проверку записанной информации и обязательную термотренировку запрограммированной микросхемы при определенной температуре (около 100°).

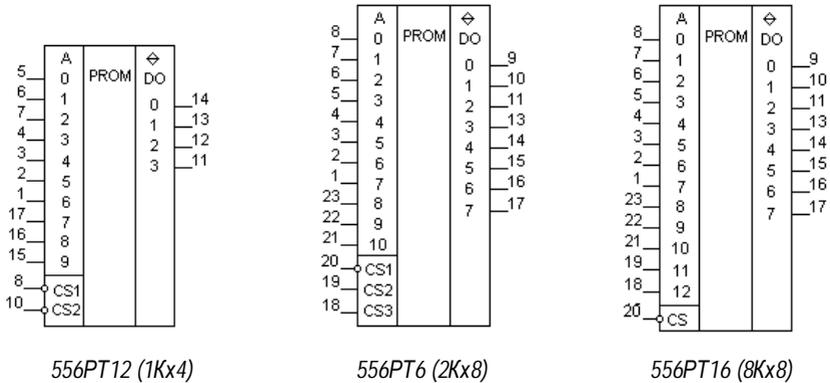


Рис. 12.13. Микросхемы OTP ROM

Микросхемы EPROM

Микросхемы EPROM способны к многократному (от 100 до 10 млн.) перепрограммированию. В качестве элементов памяти в таких микросхемах используются бистабильные транзисторы со структурой МНОП (металл – нитрид – окисел – полупроводник) или ЛИЗМОП (транзисторы n-МОП с использованием лавинной инжекции заряда).

В транзисторах МНОП диэлектрик под затвором транзистора состоит из двух слоев. При высоком напряжении на затворе (около 30В) происходит туннельное перемещение носителей заряда через очень тонкий слой окиси кремния к границе двух диэлектриков. На границе диэлектриков образуется заряженный слой, который влияет на пороговое напряжение МОП-транзистора.

У транзисторов ЛИЗМОП между управляющим затвором и подложкой расположен второй «плавающий» затвор, окруженный диэлектриком. Хранимая информация определяется зарядом на плавающем затворе.

Эквивалентная схема элемента памяти ППЗУ показана на рис. 12.14. Если заряд на плавающем затворе отсутствует, то пороговое напряжение транзистора близко к нулю. При отрицательном заряде на плавающем затворе пороговое напряжение полевого транзистора увеличивается до нескольких вольт.

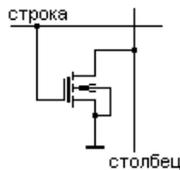


Рис. 12.14. Элемент памяти EPROM

Записанная в элемент памяти информация может храниться годами, поскольку путей для утечки заряда практически нет.

В режиме считывания на шину строки приходит единичный сигнал. В выбранной строке транзистор элемента памяти может быть закрыт или открыт в зависимости от заряда на плавающем затворе. По разрядной шине столбца появится будет протекать ток при хранении логического нуля (транзистор откроется) или же ток будет отсутствовать при хранении логической единицы (транзистор заперт). Ток столбца воспринимается и усиливается усилителем считывания.

В режиме записи в ячейку логической единицы на шину выбранного столбца подается высокое (порядка 10 В) напряжение, на шину выбранной строки подается ещё более высокое напряжение. Поскольку напряжение на затворе превышает напряжение на стоке транзистора то в пленке окисла создается высокая напряженность электрического поля и электроны, инжестрированные в окисел, дрейфуют к плавающему затвору и накапливаются в нем. Ток через диэлектрик очень мал, поэтому время программирования весьма велико (около 1 мс).

Стирание информации (удаление электронов с плавающего затвора) производится облучением кристалла ультрафиолетовым светом, для чего в корпусе микросхемы предусматривается окно с кварцевым стеклом. Под действием света электроны приобретают энергию, достаточную для перехода с плавающего затвора в окисел. Далее они дрейфуют в подложку. Стирание микросхемы происходит в течении нескольких минут. Время стирания зависит от расстояния до источника облучения, его мощности и объема микросхемы (более емкие микросхемы стираются быстрее).

Наиболее популярные микросхемы EPROM имеют восьмибитную организацию и выходы с тремя состояниями. Производством микросхем EEPROM занимается большинство ведущих производителей микроэлектроники, таких как Intel, AMD, Siemens, Motorola, Atmel. Отечественные микросхемы сосредоточены, преимущественно, в серии 573. В таблице 12.2, для примера, приведены данные ряда отечественных микросхем EPROM и популярные EPROM фирмы Atmel.

Таблица 12.2. Примеры микросхем EPROM

Отечественные микросхемы	573РФ2 2Кx8	573РФ3 4Кx16	573РФ4 8Кx8	573РФ7 32Кx8	573РФ9 128Кx8
Микросхемы Atmel	AT27BV512 64Кx8	AT27BV010 128Кx8	AT27BV020 256Кx8	AT27BV040 512Кx8	AT27C080 1Мx8

Несколько микросхем EPROM изображено на рис. 12.15. Микросхемы, как правило, имеют вход разрешения доступа #CS (*Chip Select*), вход управления выходными буферными схемами #EO (*Enable Output*), вход программирования #PR (*Programming*), вход для подачи программирующего напряжения питания V_{pp} ., адресные входы $A_0, A_1 \dots$ (*Address*) и двунаправленные выходы для чтения данных и их ввода в режиме программирования $DIO_0, DIO_1 \dots$ (*Data Input/Output*)

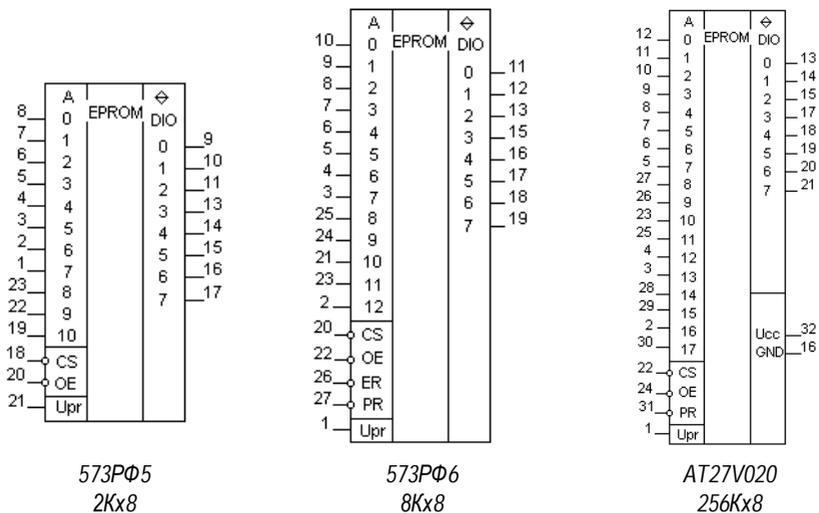


Рис. 12.15. Микросхемы EPROM

При обращении для чтения на адресные входы микросхем $A0...A_n$ подается код адреса, на входы $\#CE$ и $\#OE$ – напряжение низкого уровня. Напряжение на входе V_{pp} устанавливается равным напряжению питания V_{cc} . Считываемые данные появляются на выходах $DIO0...DIO7$. Типовые временные диаграммы сигналов в режиме чтения приведены на рис. 12.16.

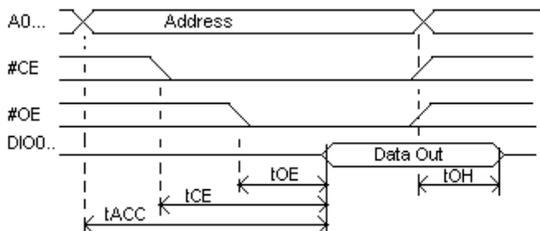
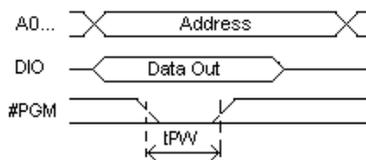


Рис. 12.16. Временные диаграммы чтения микросхем EPROM

У современных микросхем EPROM время t_{ACC} и t_{CE} порядка 50-100 нс, t_{OE} – от 20 до 50 нс, t_{OH} равно 5-7 нс.

Режимы программирования у микросхем разных производителей могут быть различны. Например, у микросхем Atmel при обращении для записи (программирование) на вход V_{cc} подается напряжение +6,5 В, на вход V_{pp} – напряжение +13 В. Запись производится при низком уровне напряжения на входе $\#CS$ и $\#PGM$ и высоком уровне напряжения на входе $\#OE$.

Типовые временные диаграммы сигналов в цикле программирования показаны на рис. 12.17.



#CS = 0, #OE = 1

Рис. 12.17. Временные диаграммы программирования EPROM

Длительность импульсов программирования t_{PW} у современных микросхем находится в пределах 50 – 100 мкс.

Микросхемы EEPROM

Микросхемы с электрическим стиранием допускают многократное (не менее 10 тыс. циклов) перепрограммирование без изъятия из устройства, в котором они работают. Гарантийный срок хранения информации - несколько лет. В структуре таких микросхем предусмотрены коммутаторы режимов и формирователи импульсов требуемой амплитуды и длительности для работы в режимах программирования и стирания. Распространенные микросхемы имеют восьмибитную организацию, выходы с тремя состояниями. Параметры некоторых отечественных и зарубежных микросхем приведены в таблице 12.3.

Таблица 12.3.

Микросхемы EEPROM

Тип микросхемы	Изготовитель	Технология	Объем памяти
558PP2	Россия	МНОП	2Kx8
558PP3	Россия	МНОП	8Kx8
1601PP3	Россия	p-МНОП	2Kx8
AT28LV64	Atmel	CMOS	8Kx8
AT28LV256	Atmel	КМОП	32Kx8
AT28LV010	Atmel	КМОП	128Kx8
AT28C040	Atmel	CMOS	512Kx8

Микросхемы EEPROM обычно имеют следующие входы и выходы (рис. 12.18):

- входы V_{CC} и GND для подключения к источнику напряжения питания;
- адресные входы $A_0 \dots A_n$;
- входы-выходы данных $DIO_0 \dots DIO_7$,
- управляющие входы: выбор микросхемы CS (*Chip Select*), разрешение чтения #OE (*Output Enable*), разрешение записи #WE (*Write Enable*).

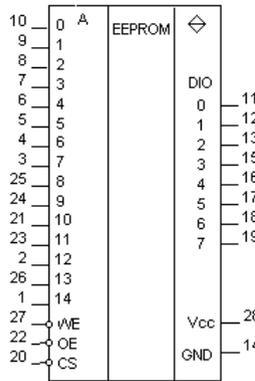


Рис. 12.18. Микросхема AT28LV64B

При обращении для чтения на адресные входы A0... подается код адреса, на входы #CS и #OE – сигнал логического нуля, на вход #WE – высокий уровень напряжения. Считываемый байт данных появляется на выходах DIO. Временные диаграммы сигналов при чтении EEPROM аналогичны рис. 2.16. У современных микросхем: tACC 60-250 нс, tOE – 30-100 нс.

При обращении для записи (*Byte Write*) на входы A0...An подается код адреса (Address), на входы #CS и #WE - низкий уровень напряжения, на вход #OE – высокий уровень напряжения, а на выходы DIO0...DIO7 – байт данных для записи. Код адреса запоминается во внутреннем регистре адреса и блоке управления по отрицательному фронту более позднего сигнала на входе #CS или на входе #WE. По этому же фронту в блоке управления запускается внутренний таймер, управляющий процедурой записи. Байт данных запоминается во внутреннем регистре данных в блоке управления по положительному фронту сигнала на входе #CS или на входе #WE (который появится раньше). После запоминания байта данных в микросхеме под управлением внутреннего таймера осуществляется перепись байта данных из регистра в блоке управления в запоминающее устройство по выбранному адресу. Временные диаграммы сигналов при обращении для записи приведены на рис. 12.19.

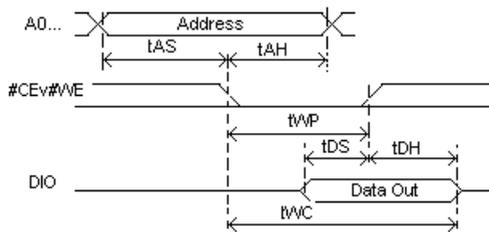


Рис. 12.19. Временные диаграммы записи EEPROM

Величины tAS и tDH у микросхем разных типов имеют значения от 0 до 10 нс. Значение tAH и tDS находятся в пределах 50-100 нс, tWP от 100 до 1000 нс.

В течении цикла записи новое обращение к памяти для записи или чтения невозможно, но шины адреса и данных микросхемы могут быть задействованы другими устройствами.

Для сокращения суммарного времени, затрачиваемого на запись группы байтов, в микросхемах многих типов реализуется режим страничной записи (Rage Write). В таком режиме группа байтов данных и их адресов вводится в микросхему последовательно, байт за байтом, и запоминается в регистровом запоминающем устройстве блока управления. Старшие разряды кода адреса, одинаковые для всех микросхем в группе, задают номер страницы памяти, а младшие разряды – определяют адрес байта в пределах страницы. Число регистров в регистровом запоминающем устройстве определяет размер страницы памяти. Количество страниц памяти в современных микросхемах в пределах 64 – 256. Например, микросхема AT28LV64B имеет 64 страницы памяти. Если выполнена запись во все регистры регистрового запоминающего устройства или в течение определенного интервала времени (tBLC) после записи байта не произошла запись очередного байта данных, в микросхеме выполняется цикл записи, в котором введенные ранее данные переписываются в память по заданным адресам.

Для предотвращения случайной записи в память при включении напряжения питания и появления помех в шинах управления в микросхемах EEPROM обычно предусматриваются аппаратные средства защиты данных. Например, все микросхемы EEPROM фирмы Atmel имеют следующие средства защиты (Hardware Protection):

- запись не выполняется, если величина напряжения питания не достигла установленного порога;
- запись не выполняется в течении некоторого интервала времени после достижения напряжением питания установленного порога;
- запись не выполняется, если хотя бы один из трех сигналов на входах управления #CS, #WE, #OE не имеет требуемого значения;
- запись не выполняется, если длительность импульсов на входах #CS или #WE меньше некоторой пороговой величины.

Кроме аппаратных средств защиты в микросхемах с режимом страничной записи используются и программные средства (*Software Data Protection*). Причем режим программной защиты может быть как введен, так и отменен. Например, в микросхеме AT28C256 фирмы Atmel для установки режима программной защиты необходимо по адресам 0x1555, 0x0AAA, 0x1555 ввести последовательность байтов: 0xAA, 0x55, 0xA0. После этого, перед любой записью данных также необходимо вводить эту последовательность байтов.

Многие микросхемы EEPROM имеют режим аппаратного стирания памяти (*Chip Clear*). Так, у микросхем типа AT28C16, AT28C17, AT28C64/X фирмы Atmel стирание памяти происходит при #CS=0, напряжении +12В на входе #OE и подаче на вход #WE нулевого сигнала в течении 10 мс.

В микросхемах некоторых типов имеется возможность программного стирания памяти (*Optional Chip Erase Mode*). Например, в микросхемах AT28C265, AT28C010, AT28C040, AT28C256 стирание происходит после записи по адресам 0x5555, 0x2AAA, 0x5555, 0x5555, 0x2AAA, 0x5555 последовательности байтов 0xAA, 0x55, 0x80, 0xAA, 0x55, 0x10. Стирание происходит за 20 мс.

Флэш-память

Флэш-память относится к классу EEPROM, но использует особую технологию построения запоминающих ячеек. В отличие от EEPROM, в микросхемах Flash-памяти процесс записи всегда сопровождается предварительным стиранием ячеек сектора данных или всей микросхемы. Первые микросхемы флэш-памяти были предложены фирмой Intel в 1988 году и с тех пор в различных видах выпускаются различными фирмами.

Они отличаются по архитектуре, объему памяти и напряжениям питания. Терминология в этой, сравнительно новой области, полностью не устоялась. Поэтому в качестве примера рассмотрим классификацию микросхем, предлагаемую фирмами Atmel и Intel.

Микросхемы, как правило, имеют выводы (рис. 12.20 и 12.21):

- входы Vcc и GND для подключения к источнику питания;
- адресные входы A0...An;
- входы/выходы данных DIO0...DIO7;
- управляющие входы: выбор микросхемы #CS (Chip Select), разрешение чтения #OE (Output Enable), разрешение записи #WE (Write Enable).

Некоторые микросхемы имеют дополнительный управляющий вход #RESET, используемый для перевода их в пассивное состояние.

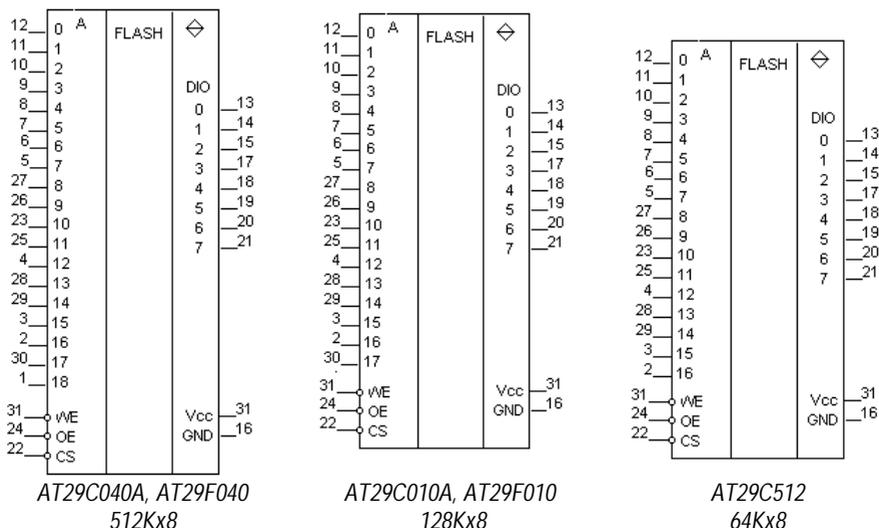


Рис. 12.20. Микросхемы флэш-памяти фирмы Atmel

При чтении данных из микросхем на адресные входы A0...An подается код адреса (Address), на входы #CS и #OE – нулевой сигнал, на вход #WE – единичный. Считываемые данные появляются на выводах DIO0...DIO7. Временные диаграммы сигналов при чтении совпадают с сигналами чтения микросхем EPROM (рис. 12.16). Время tACC равно 70-250 нс, время tOE – от 35 до 120 нс.

Запись данных в микросхемы производится по-разному, в зависимости от её типа.

В микросхемах типа AT29 реализован секторный способ записи. В них блок ячейек памяти разделен на секторы равной емкости. Число секторов и емкость сектора ряда микросхем приведены в таблице.12.4.

Таблица 12.4.

Микросхемы flash-памяти фирмы Atmel с секторным способом записи

Тип микросхемы	Емкость	К-во секторов	Емкость сектора	Емкость блока
AT29C256	32Kx8	512	64x8	
AT29C512	64Kx8	512	128x8	

AT29C010A	128Kx8	1024	128x8	8Kx8
AT29C020	256Kx8	1024	256x8	8Kx8
AT29C040A	512Kx8	2048	256x8	16Kx8

Блок управления таких микросхем содержит регистровое запоминающее устройство, емкость которых равна емкости сектора.

Процедура записи данных в микросхему разделена на два этапа. На первом этапе данные и их адреса вводятся в микросхему и запоминаются в регистровом запоминающем устройстве. В коде адреса старшие разряды задают номер сектора, одинаковый для всех кодов в группе. Младшие разряды кода адреса определяют регистр в регистровом запоминающем устройстве и соответствующую ему ячейку в заданном секторе памяти. Запись кодов в регистровое запоминающее устройство может выполняться в произвольном порядке и не во все регистры.

На втором этапе выполняется цикл записи, в течение которого осуществляется стирание всех ячеек заданного сектора и перепись данных из всех регистров из регистрового запоминающего устройства в соответствующие ячейки выбранного сектора памяти. Переход ко второму этапу осуществляется автоматически, если выполнена запись во все регистры запоминающего устройства или, если после ввода данных в течение интервала времени $t_{BLC} = 150$ мкс не вводится очередной байт данных. Временные диаграммы сигналов управления на первом и втором этапах записи данных совпадают с временными диаграммами записи в микросхемы EEPROM при страничной записи (рис. 12.19). Величины t_{AS} и t_{DH} при этом равны 0-10 нс, $t_{AH}=50-100$ нс, $t_{DS}=35-100$ нс, $t_{WP}=70-200$ нс.

В микросхемах памяти семейства AT49 реализован побайтный способ записи с предварительным стиранием всех ячеек памяти. Процедура стирания памяти (*Chip Erase*) запускается после ввода в микросхему последовательности из шести байтов: 0xAA, 0x55, 0x80, 0xAA, 0x55, 0x10 по адресам: 0x5555, 0x2AAA, 0x5555, 0x5555, 0x2AAA, 0x5555. Стирание происходит за 10 с.

После завершения процедуры стирания выполняется побайтная запись. Перед записью каждого байта данных в микросхему вводится последовательность из трех байтов: 0xAA, 0x55, 0x80, 0xA0 по адресам: 0x5555, 0x2AAA, 0x5555. Затем по требуемому адресу вводится байт данных. После ввода байта данных в микросхеме выполняется цикл записи, длительностью 50 мкс.

Кроме программных средств защиты памяти от случайной записи во всех микросхемах flash-памяти фирмы Atmel используются аппаратные средства защиты (*Hardware Data Protection*) от разрушения данных при включении напряжения питания и появления помех в сигналах управления. Они аналогичны, описанным ранее, средствам аппаратной защиты микросхем EEPROM.

В микросхемах flash-памяти обычно имеются блоки памяти BB (*Boot Block*)? которые особым образом защищаются от чтения и записи. В некоторых микросхемах Atmel семейства AT29 таких блоков два BB1 и BB2. Блоки имеют одинаковую емкость, обычно 8Kx8. Адреса для обращения к ячейкам в первом блоке BB1 расположены в начальной части адресного пространства. Адреса для обращения к ячейкам во втором блоке BB2 расположены в конечной части адресного пространства. Для введения режима защиты необходимо ввести в микросхему последовательность из шести заданных байтов по определенным адресам и после этого ввести седьмой байт 0x00 по адресу 0x0B для защиты блока BB1 или по адресу 0x111...B для защиты блока BB2.

В микросхеме flash-памяти обычно при изготовлении записывается код идентификации, подтверждающий тип микросхемы (*device code*) и факт её изготовления

конкретной фирмой (*manufacturer code*). Специальная последовательность из трех кодов, записанных по определенным адресам, позволяет прочесть эти коды.

Фирма Intel в настоящее время развивает три основных типа микросхем флэш-памяти:

Микросхемы со структурой *Bulk Erase (BE)*, стираемые целиком и программируемые побайтно.

Микросхемы со структурой *Boot Block (BB)*, в которых весь массив ячеек памяти разделен на блоки одинакового размера, стираемые независимо. Один из блоков – привилегированный блок – имеет дополнительные аппаратные средства защиты от стирания и записи.

Микросхемы с симметричной архитектурой *SA (Symmetrical Architecture)*, называемые также микросхемами со структурой *Flash File*, в которых массив ячеек памяти разделен на несколько независимо стираемых блоков одинакового размера.

В таблице 12.5 приведены сравнительные данные на некоторые микросхемы флэш-памяти, выпускаемые фирмой Intel.

Таблица 12.5.
Микросхемы flash-памяти фирмы Intel

Тип микросхемы	Организация
28F256	32Kx8 BE
28F512	64Kx8 BE
28F010	128Kx8 BE
28F020	256Kx8 BE
28F001	128Kx8 BB
28F002	256Kx8 BB
28F004	512Kx8 BB
28F008	1024Kx8 BB
28F200	256Kx8, 128Kx16 BB
28F400	512Kx8, 256Kx16 BB
28F800	1024Kx8, 512Kx16 BB
28F002	256Kx8 SA
28F004	512Kx8 SA
28F008	1024Kx8 SA

Несколько микросхем из таблицы изображено на рис. 12.21.

Время доступа микросхем *Bulk Erase* лежит в пределах 65...200 нс. Гарантированное число циклов стирания-программирования от 10000 до 100000.

Независимо от структуры обращение к микросхемам разрешается при низком уровне сигнала *CE (Chip Enable)*. Высокий уровень сигнала *CE* переводит схемы в режим пониженного энергопотребления.

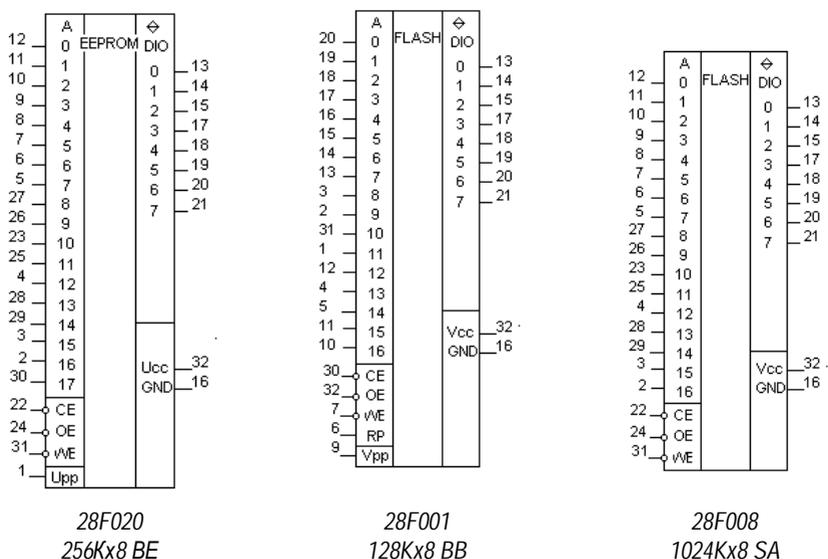


Рис. 12.21. Микросхемы флэш-памяти фирмы Intel

Вход *OE* (*Output Enable*) разрешает работу выходных буферов схемы. При *OE=0*, *CE=0* разрешается чтение данных из микросхемы. При *OE=1* на выходах схемы устанавливается высокоимпеданное состояние.

Сигнал *WE* (*Write Enable*) разрешает запись (при *CE=0*) и одновременно переводит выходные буферы в высокоимпеданное состояние независимо от сигнала *OE*.

У микросхем *BB* дополнительный управляющий сигнал *RP=0* (*Reset/Power Down*) переводит микросхему в режим «глубокого сна» с потреблением тока в доли микроампера. Перевод сигнала *RP* в единичное состояние переводит микросхему в режим чтения данных. Поддача на вход *RP* напряжения +12В разрешает программирование даже защищенного *Boot-блока*.

Стирание и программирование микросхем *BE* и *BB* возможны только при подаче на вход *Vpp* напряжения 12В. В этом случае во внутренний регистр команд микросхемы по сигналу *WE* могут быть записаны определенные команды для управления микросхемой.

Команда *Read Memory* (код команды 00h) переводит микросхему в режим чтения.

Команда *Read ID* (код 90h) – команда чтения идентификаторов. В последующих циклах чтения по адресу 0 считывается идентификатор производителя *M_Id* (*Manufacturer Identifier*), а по адресу 1 – идентификатор устройства *D_Id* (*Device Identifier*). Для микросхем 28F256, 28F512, 28F010, 28F020 они соответственно равны B9h, B8h, B4h, BDh.

Команда *Set-up Erase/Erase* (код 20h) – подготовка к стиранию и стирание микросхемы. Внутренний цикл стирания начинается по переднему фронту сигнала *WE*.

Команда *Erase Verify* (код A0h) – верификация (проверка) стирания. Выполняется последовательно для всех ячеек микросхемы. Если результат считывания отличается от FFh, то производится повторное стирание (длительностью 10 мс) Если количество повторов стирания превышает 3000, фиксируется ошибка стирания и микросхема признается негодной.

Команда *Set-up Program/Program* (код 40h) – подготовка к программированию и программирование. Выполняется аналогично стиранию, но на микросхему передаются адрес и данные программируемой ячейки.

Команда *Program Verify* (код C0h) – верификация программирования. В случае несоответствия результата выполняется повторное программирование (до 25 раз для каждой ячейки). Далее фиксируется отказ микросхемы.

Команда *Reset* (код FFh) – сброс. Команда прерывает программирование и стирание. После неё необходима подача другой команды.

При включении питания внутренний регистр команд микросхемы обнуляется, что соответствует команде *Read Memory*. При подаче на вход Vpp низкого напряжения (менее 6,5 В) стирание и программирование невозможны и микросхема ведет себя как обычное ПЗУ типа EPROM.

В микросхемах флэш-памяти второго поколения ячейки памяти группируются в блоки, допускающие независимое стирание. Операция стирания каждого блока может прерываться для считывания данных других блоков. По способу разбиения на блоки различают микросхемы *Boot Block* и *Flash File*.

В микросхемах *Boot Block* массив ячеек памяти разделен на несколько блоков разного размера, стираемых независимо. Один из блоков (*Boot Block – привилегированный блок*) имеет дополнительные аппаратные средства защиты от стирания и записи. Микросхемы имеют однобайтную или переключаемую 8/16 бит разрядность. В микросхемах *Flash File* массив ячеек разделен на несколько равноправных независимо стираемых блоков одинакового размера. Их второе название – микросхемы с симметричной архитектурой (*SA – Symmetrical Architecture*). Параметры ряда микросхем приведены в табл. 10.5

ЛИТЕРАТУРА

1. Угрюмов Е.П. Цифровая схемотехника.- СПб.: БХВ – Санкт-Петербург, 2000.- 528 с.
2. Аваев Н.А. и др. Основы микросхемотехники: Учебное пособие для вузов/ Н.А.Аваев, Ю.Е.Наумов, В.Т.Фролкин.- М.: Радио и связь, 1991. - 288 с.
3. Степаненко И.П. Основы микросхемотехники: Учеб пособие для вузов.- М.: Лаборатория базовых знаний, 2001.- 488 с.
4. Схемотехника ЭВМ: Учебник для ВУЗов/ Под ред. Г.Н.Соловьева.- М.: Высш. шк., 1985.- 391 с.
5. Горбачев Г.Н., Чаплыгин Е.Е. Промышленная электроника: Учебник для вузов/ Под ред. В.А.Лабунцова.- М.: Энергоатомиздат, 1988.- 320 с.
6. Шило В.Л. Популярные цифровые микросхемы: Справочник.- М.: Радио и связь, 1989.- 352 с.
7. Титце У., Шенк К. Полупроводниковая схемотехника: Справочное руководство.- М., Мир, 1982.- 512 с.
8. Опадчий Ю.Ф. и др. Аналоговая и цифровая электроника: Учебник для ВУЗов.- М.: Радио и связь, 1996.- 768 с.
9. Аналоговые и цифровые интегральные микросхемы. Справочное пособие/ С.В.Якубовский, Н.А.Барканов, Л.И.Нисельсон и др.- М.: Радио и связь, 1985.- 432 с.
10. Янсен Й. Курс цифровой электроники: в 4-х т. - М.: Мир, 1987.- 334 с.
11. Микросхемы памяти. ЦАП и АЦП: Справочник/ О.Н.Лебедев, А.-Й.К. Марцинкявичус, Э.-А.К.Багданскис и др.; М.: КубК-а, 1996.- 384 с.
12. Шипулин С.Н., Храпов В.Ю. Основные тенденции развития ПЛИС // Электронные компоненты. — 1996. — № 3-4. — С. 26.

13. Стешенко В. Школа разработки аппаратуры цифровой обработки сигналов на ПЛИС.// Chip News
14. Ермаков А.Е., Крюков В.П., Назаров Н.Б., Петров Л.Н. Базовые матричные кристаллы типа ТТЛШ серии К1548. — Электронная промышленность. — 1991. — Вып. 7. — 105 с.
15. Домрачев В.Г., Мальцев П.П., Новаченко И.В., Пономарев С.Н. Базовые матричные кристаллы и матричные БИС. — М.: Энергоиздат. — 1992. — 225 с.
16. Гольдшер А., Юргаев Б.. Цифровой базовый матричный кристалл К 1589ХМ1



17. В. В. Гребнев.
Микросхемы энергонезависимой памяти фирмы ATMEL.- С.Петербург: ЭФО.- 1997.



18. Антонов А. П., Мелехин В. Ф., Филиппов А. С.. Обзор элементной базы фирмы ALTERA.- С.Петербург: ЭФО.- 1999
19. Цифровые интегральные микросхемы: Справочник/ Мальцев П.П., Долидзе Н.С., Критенко М.И. и др.- М.: Радио и связь, 1994.- 240 с.
20. Стешенко В.Б. ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов. – ДОДЕКА, 2000.- 128 с.
21. Гребнев В.В. Микросхемы энергонезависимой памяти фирмы Atmel.- С.Пб.: ЭФО, 1997.- 64 с.

ОГЛАВЛЕНИЕ

ЦИФРОВЫЕ ЭЛЕМЕНТЫ СИСТЕМ АВТОМАТИКИ

Учебное пособие

